COMMODORE user

Incorporating Vic Computing

Volume 1 Issue 1 October 1983 Price 85p

Happy birthday to us!



Welcome to our first issue

Butterfield serenades the birth

Old friends gather round: Todd, Tommy, Punter
• Readers write • Reviews: 64 Sound, Graphic Packages, Epson, books, 64 WP

Joystick control
Victuals
The Graphics Primer
Intro to assembler


EXPLORE
NEW FRONTIERS OF FUN
WITH
Audiogenic
From vintage classics like "AMOK" and "ALIEN BLITZ" to the latest classics like "KAKTUS" and "MOTOR MANIA".
From Machine Code Arcade Action through Graphical Adventure to Mind Bending Board Games - WE'VE GOT THE LOT!
THE BIGGEST RANGE EVER FOR VIC 20 and 64 - NOW ALSO DRAGON AND SPECTRUM
WRITE OR PHONE FOR FREE COLOUR CATALOGUE!
AUDIOGENIC, P.O. BOX 88, READING, BERKS. (0734) 586334
MORE THAN JUST A GAME...

COMMODORE user

Contents

Editor
Dennis Jarrett
Writer
Bohdan Buciak
Editorial Assistant
Nicky Chapman
Business Manager
Annabel Hunt
Reader Services Manager
Hywel David
Subscriptions
Janet Crawford
Art/Production Editor
Paul Hulme

Production
Maggie Underwood, Michael Pattenden
Advertising
Cathy Lane, Roland Richardson
Heads of Vicwear
Ned and Vicki Staples

Front Cover
Photo: Rogue Images
Cake
G J Pearce Ltd
Hertford (by appointment)

Telephones:

Editorial	01-241 2448
Advertising	01-241 2417
Production	01-241 2493
Administration	01-241 2354
Subscriptions and Back Issues	0732-351216

Commodore User is published twelve times a year by the Paradox Group, The Metropolitan, Enfield Road, London N1 5AZ.
Subscriptions and back issues are handled by B.K.T. Subscription Services, Dowgate Works, Douglas Road, Tonbridge, Kent TN9 2TS; Telephone 0732-351216
Annual subscription (UK) £12.00, (Eire) £IRL17.00, (EUROPE) £18.00, (REST OF WORLD) £32.00.

Cheques and postal orders should be made payable to **Vic Computing Ltd.**

Printed by Garrod & Lofthouse Ltd, Crawley; Phototypesetting by Carlinpoint Ltd, London WC2 and PDQ Ltd London N10; Reproduction by Elan Litho Ltd, London EC2.

Material is accepted on an 'all rights' basis unless otherwise agreed. Enclose SAE if you want the return of original manuscripts and photographs.

Please do not send listings: we only accept programs in cassette form.

Publisher's guarantee: There are no mistakes in this magazine except this one.

LLAMASOFT!!

MORE THAN JUST A PRETTY PACKET!

NEW NEW NEW NEW

HOVER BOVVER –
THE LATEST AND GREATEST FOR THE
CBM 64 FROM LLAMASOFT

HOVER BOVVER



HOVER BOVVER - a totally original arcade game for C64 featuring outstanding graphics and a sound track created by a professional Piano Wizard. Gordon Bennet has borrowed his neighbour's Air-Mo lawnmower. Mow your way through as many of the 16 lawns as you can before the pursuing neighbour retrieves his mower. Set your dog onto the neighbour to help you out of tight spots and don't annoy the Gardener. Try not to plough through the neat flower beds or overheat your mower!

LLAMASOFT!! Computer Software

Dept CU1, 49 Mount Pleasant, Tadley,
Basingstoke, Hants. RG26 6BN
☎ Tadley 07356 4478
TRADE ENQUIRIES WELCOME



Simply File: DBMS for 64

Simple Software is quite well known for its useful low-cost high-quality packages for the Commodore Pet/CBM line. The good news is that the company is getting interested in the 64; it has just introduced Simply File for the CBM 64 with disk drive.

We'll be reviewing it, but here's a taster. For a start, Simply File looks like a very versatile records management system for a wide range of educational and business use. Records can be indexed, searched and grouped by many criteria, and the system can also be used in conjunction with another good piece of work from Simple Software – the low-cost Simply Write word processor (also under review, as it happens).

The database program comes as eleven modules, loaded automatically as needed. It will print summaries of selected records, print labels, carry out calculations on all or selected groups of records, and even output information as sequential files for use with other programs; it may be used for single or dual disks and outputs to any Commodore or non-Commodore printer (IEEE, RS-232 or Centronics). It allows about 400 records of 254 characters on a single 1541 disk (with programs on the same floppy), or more in proportion for smaller records.

The price? Simply File costs £60 plus VAT from dealers, or direct from Simple Software Ltd; more info on Brighton 504879.

Softsel's hits

The software distributor Softsel arrived in the UK quite noisily in the Summer. It's a distributor, taking other people's products (a wide variety of programs on a lot of different computers) and supplying retailers (over 4,000, most in the States but across 30 countries internationally as well).

Softsel publishes a promotional 'hot list' for retailers showing its best-sellers. The last one we saw was for the week of 8 August, and there's some interesting offerings on it. Sadly they aren't split by machine, and some of the packages run on several micros. Top seller is something called *Zaxxon*, but here's the Commodore-related content from the games top 50 (running on 64 unless indicated):

2	Zork 1	great *adventure* game from a US specialist called Infocom. We thought it was IBM-only, but it seems to be on the 64 too.
3	Frogger	from Sierra On-line, whoever they are.
4	Deadline	another excellent Infocom adventur with a difference – it's an Agatha Christie world, advanced *Cluedo* for computers Good for the IBM PC, must be great on the 64.
5	Witness	a newish one from Infocom; we haven't seen it.
9	Zork II	and it's Infocom again This is the advanced, improved, costlier version.
10	Jump Man	who he? From Epyx (who they?)
12	Temple of Asphai	another Epyx one, though we have seen this adventure on other computers. It's ok.
13	River Rescue	hooray for Britain! Thorn-EMI's Vic cartridge, the best of their collection.
14	Zork III	... and again. Didn't know there *was* a Zork III!
16	Suspended	yet another Infocom entry.
18	Gridrunner	the ace Llamasoft game for Vic and 64.
19	Starcross	Infocom once more.
20	Submarine Commander	Thorn-EMI's Vic and 64 cartridges.
24	AE	don't know it. But it's by Broderbund, who did the very excellent *Choplifter*.
29	Sargon	Vic chess – maybe not the best, but certainly the biggest seller.
32	Repton	by Sirius, but we don't know it.
36	Sword of Fargoal	who he? Epyx again, Vic and 64.
49	Crossfire	Sierra On-line, another one we don't know.

Clean it up

According to B&R Electrical Products, a lot of sensitive electronic equipment is at risk from sudden spikes or surges on the mains – office computers and word processors, electronic typewriters, home computers, videos, cash registers, hi-fi equipment, instruments and controllers, and so on.

Mains-borne interference like that can be generated by switches, fluorescent lights, motorised equipment, air conditioning, vacuum cleaners, washing machines, printers ... These everyday surges can be in excess of 3,000V; thunderstorms can easily generate over 10,000V on the mains.

You won't then be too amazed to learn that "for such sensitive and valuable equipment B&R recommends fitting its new PowerCleaner surge suppressor plug." This just replaces an existing 13 Amp mains plug, yet it can provide a dramatic reduction of up to 99% in equipment damage or disruption.

Most modern electronic gear is designed to withstand surges to 600 to 700V. Anything greater than this can actually cause permanent damage, though a one-time loss of data or program is more likely. The PowerCleaner plug effectively limits surges to 625V, a level at which B&R says "tests show a 99 per cent reduction in equipment damage from spurious surges".

Priced not unreasonably at £8.65 plus VAT, PowerCleaner is available from B&R stockists: further info on 0279 34561.

First 64 assembler?

Supersoft reckons it's first with a 6502/6510 assembler for the 64 – as opposed to machine code monitors, that is. "Mikro 64 makes programming in assembler almost as easy as programming in Basic!" exclaims Supersoft.

The package is an uprated version of Supersoft's successful Mikro for the Pet. The 64 implementation adds some goodies, like merging of source files to create machine-code programs of up to 12K. You can write, save, load, verify and delete just like Basic ... except that there're some helpful built-in commands like AUTO, FIND and DELETE.

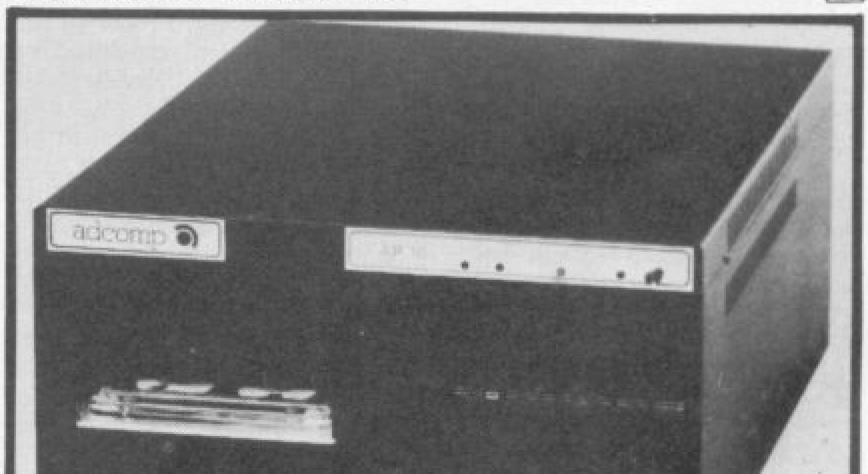
It comes as a cartridge for £50 plus VAT, and with it there is a built-in machine code monitor and disassembler. Supersoft has a number of dealers, but try 01-861 1166 if there's any difficulty.

On the other hand, Kobra has PAL – "a Personal Assembly Language" for the 64. Sounds like an assembler to us ...

It needs 4KB and a 1541 disk. Kobra says it has been designed to fit in "as far as possible" with the Basic environment, so the transition from Basic to Assembler programming should be relatively easy.

A two-pass assembler, PAL reads each statement on the first pass to determine the number of bytes it requires before generating the necessary machine code during the second pass. Capabilities include full labelling, allocation of Opcodes and Operands, and a range of addressing modes. But Kobra thinks the major plus for PAL resides in the expression evaluator. The assembler allows the use of complex expressions involving a wide set of operators and operand formats: operations such as addition and subtraction are valid. Terms can be decimal, hexadecimal, binary, ASCII, labels or expressions.

PAL costs £69 plus VAT and is available from Kobra (Farm Road, Henley on Thames, Oxon) or the nationwide Commodore dealer network.



In the picture is a new mini-Winchester disk (uses a 5¼in non-removable hard disk) for the CBM 8000. It stores 18 or 36 megabytes, has an integral tape streamer for backup, and includes enough intelligence to provide for multi-user operation – up to 12 8000's can attach to it, up to 40 relative files can be opened at the same time, and file access protection is built in via disk status procedures. It hails from IMI Computing – details from 021-356 4848.



Guilding the lily? Gosh ...

If you thought the software industry had no conscience and didn't give a damn about its customers, you could be right. Or maybe not. The impressively-named Guild of Software Houses has just been formed by a group of leading small-computer software sellers including Bug-Byte, New Generation, Quicksilva, Salamander, Softek and Virgin Games.

This formidable line-up has produced a Customers' Charter that guarantees the kind of service you've always dreamed about ... or should be getting anyway. The Charter is really a code of practice which all members must adhere to. It guarantees prompt despatch of orders and replacement of faulty merchandise free of charge. Members undertake to maintain good standards of customer service and deal with complaints thoroughly and quickly.

They've also been told to stop advertising products if they're not yet available; this practice is particularly frustrating for the eager-to-buy customer.

It's not altruism all the way, though. Nick Alexander of Virgin Games, who's Chairman of the Guild, points out that "It's in the interests of software houses to be more united. And our Guild logo will become a sign of quality and service". So the punters will know where to do their shopping.

The Guild hopes to grab an even larger share of the market if customers begin to associate the logo with better value and security. But of course, it could also attract dealers who might stock more of the Guild members' products. So the Guild has produced a dealers' version of the customers' charter which makes the same guarantees and the same threat of fines and expulsion for members who fail to become paragons of virtue.

This may look like heavy-handed tactics; but Nick Alexander doesn't think so. "You need clout or the whole venture becomes meaningless. You've got to be sure that people will abide by the rules." One way of making sure is to vet software houses for membership and, if they pass the test, charge them a hefty membership fee – the annual subscription is £500.

But small concerns won't be able to fork out that kind of money, surely? "We only want software houses that have already got stature in the market. We're not after quantity of membership; 30 to 40 members would be about right." The Guild's got seven so far.

So how does the Guild vet new members? "Everybody in the Guild knows the software industry so we're aware of people's reputation. If we haven't heard of a company, we'd want to talk to them and find out how they work." Sounds more like a detective agency than a Guild.

The Guild doesn't guarantee that you'll get an enjoyable game or a useful piece of software because that's usually a matter of opinion. But it does try to cut out a lot of things that could spoil your enjoyment like low technical standards and unreliability – and the Guild says it will take up customers' complaints if they get no response from the supplier.

(Aspiring members of the Guild can contact Secretary Mike Johnson on 01-801 9172. He provides the address for the Guild, which says: "Complaints should be directed in the first instance to the supplier of the goods but if this fails to provide satisfaction, write to the Guild of Software Houses, 71 Park Lane, Tottenham, London N17 0HG enclosing any relevant correspondence and the matter will be taken up on your behalf."



Ticket to ride ...

Meanwhile the not-much-longer-established Virgin Games has just announced the first-ever computer games tour. Sounds exciting and original? Well, not quite. Virgin has converted an ageing double-decker bus, painted it in gaudy colours, and filled it with Twelve Leading Home Computers – including the Vic and the Commodore 64, of course.

The bus is going on a nationwide tour soon, helped by local radio stations which will be announcing the venues over the air. Virgin hopes to make about four stops per day, coinciding with kids' lunch breaks and school finish.

The Fun Bus, as it's brilliantly named, will be promoting the eight new computer games Virgin has just announced; that brings its total to 16. Two of the new ones are for the Vic.

'Envahi', described cryptically as "the game of the film of the book of the game", was written by a 17-year-old for joystick and 8K expansion. 'Creepers' is yet another alien-zapping game for 3K or 8K expanded Vics.

Nothing new for the Commodore 64 though. "We feel the Commodore 64 is an important machine," says Nick Alexander, Virgin Games' MD. "But we're bringing out games on the Vic for the time being. Eventually production will be dropping off in favour of the 64."

Virgin's publicity man Tony Brainsby, told us that the tour will have some educational value, but he struggled to sound convincing. "The bus is not just a stall with lots of games." Oh really? "We'll be showing kids what computers are all about." We'd like to know that, too. "They'll be able to chat to our programmers." Whoopee. He also waxes lyrical about taking computers to the people and "looking to do something for the community" – like driving a zebra-striped bus up your street.

The Fun Bus is costing Virgin about £10,000. That apparently compares favourably with the cost of a poster campaign. It's also a little more flamboyant and showbizzy – just the type of thing record companies drool about.

Instant code: *Dynatech sells program generators, programs which write programs (very clever, very easy to use if you're trying to produce a complicated file-orientated system). One of them is called Codewriter; it's been around on the Pet for a while, but now there's a 64 version too – it's Commodore-approved and costs a modest £200. It should be very good; we've requested a review copy.*

Seeing red

Yet another small software house, Visions Software Factory, has just appeared in the rough and tumble computer games market helped by snooker megapersonality Steve Davies. What's the connection? Well, Visions' best-selling game so far is predictably called *Snooker*.

Snooker is available on the Vic (with 3K expansion) and the Commodore 64. If you buy the Vic version though, don't expect to get a maximum 147 break because you only get 10 reds. With the 64 you get 10 or the full 15.

Jon Burnham, one of Vision's directors, feels that violent computer games will eventually be replaced by more realistic games, like Snooker. "Real-life situations for games haven't been explored very much by the computer world. Their advantage is that you don't have to learn the playing rules. A game title is just another title; with Snooker, you know already what the game's going to be about."

Visions is only three months old but claims to have sold 20,000 tapes already. This may be due to the large distribution it has by signing with Prism Microproducts and Lightning Records: Prism, which distributes mostly Spectrum games, has a 26 per cent stake in the company. Lightning has signed up as a wholesaler. In total, this gives Visions 8,000 possible retail outlets.

Visions is a young company, and with what looks like youthful naivety, it's pledging to change the image of the software industry by "providing quality goods with professionalism". You've heard it all before, but Visions at least is doing something practical by using chrome recording tape (which should reduce drop-outs) and by duplicating on specially-designed tape machines (rather than adapted music tape duplicators).

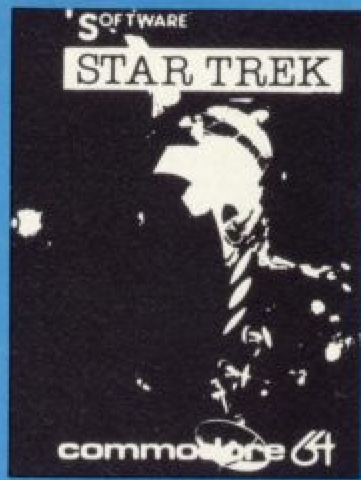
COMMODORE 64 GAMES ARE NOW AVAILABLE ON DISKETTE FOR ONLY £2.00 EXTRA

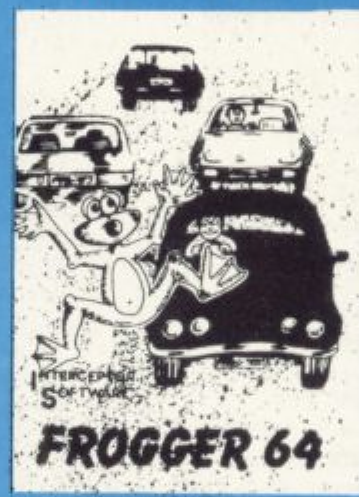
INTERCEPTOR MICRO'S PRESENTS

6 NEW GAMES FOR YOUR commodore 64

SEND S.A.E. FOR FULL LIST OF GAMES





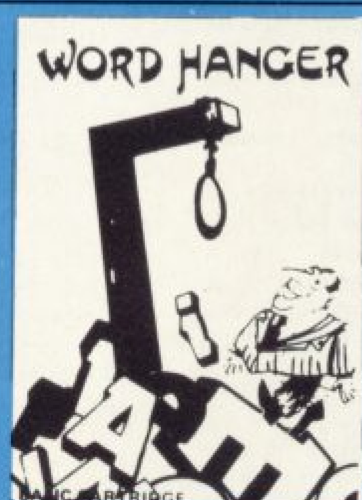




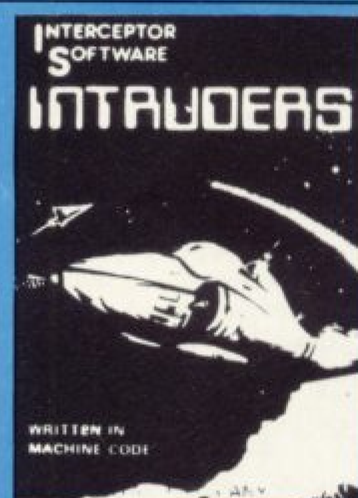








2 NEW GAMES FOR YOUR ATARI 400/800–16K



NEW GAME FOR YOUR BBC MODEL B



NEW VIC 20 GAMES

Mr, Mrs, Miss

Address

Quantity	Item		Total		Item		Total
	Scramble 64	£7.00			Galaxzions VIC-20	£5.00	
	Star Trek 64	£7.00			Fantazia VIC-20	£6.00	
	Frogger 64	£7.00			Crazy Kong VIC-20	£6.00	
	Panic 64	£7.00			Frog VIC-20	£4.00	
	Crazy Kong 64	£7.00			Alien Attack VIC-20	£6.00	
	Spriteman 64	£7.00			Vic Rescue VIC-20	£5.00	
	Fighter Pilot Atari	£7.00			Penny Slot VIC-20	£4.50	
	Wordhanger Atari	£5.00					
	Intruders BBC	£5.00					
	Jupiter Defender	£6.00			P&P	50	
					Total		

Send Cheque/P.O. with Order to: Interceptor Micro's, Lindon House, The Green, Tadley, Hants.

Postage & Packing .50p

cci Sept

STOP PRESS 1 NEW VIC 20 FALCON FIGHTER STOP PRESS STOP PRESS
ONLY £6.00

Butterfield

Commodore 64 Music:

by Jim Butterfield

"The trouble is," said my friend Andrew, "that you can't find the programs which are most wanted. For example, a program that plays 'Happy Birthday to You'. It's needed for several reasons. First, the 64 is often given as a birthday present; this should be its first program. Secondly, the computer is a member of the family, and should play a part in the celebrations..."

It didn't seem too hard a job to me. The music-playing program already existed in Basic, and Basic programs are easy to change and expand. The musically-inclined SID chip gave Basic lots of time to do extra things.

The music should be written rather slowly (people tend to sing along rather sluggishly); and the words should come up on the screen. A bouncing ball could be achieved by using sprites, but I chose a simpler approach.

Let's program along. I'll give brief notes on what's happening:

100 PRINT CHR$(147); CHR$(154); "MUSIC (BY JIM BUTTERFIELD)

Let's clear the screen, set the colour to light blue (that's normal), and place the blame.

101 READ S: IF S=0 GOTO 109
102 READ S$: GOSUB 1000
103 READ X1, Y1, X2, Y2, X3, Y3
104 GOTO 101

The above code runs through the DATA statements, printing the words of the song. The subroutine at line 1000 does the actual printing; we'll look at it later.

109 RESTORE: PRINT CHR$(19); CHR$(5)

We back up to the start of the DATA statements with the RESTORE command. Then we HOME the cursor and change our printing colour to white.

110 L1=54272: L2=54279: L3=54286
120 H1=L1+1: H2=L2+1: H3=L3+1
130 V1=L1+4: V2=L2+4: V3=L3+4
140 POKE 54296, 15

This sets the locations of the various instruments in the SID chip. L and H will be the low and high parts of the frequency (pitch) for each note: V will be the 'action' location where we strike the note. And we turn the volume up its maximum value of 15.

150 POKE V1+1,9: POKE V1+2,0
160 POKE V2+1,36: POKE V2+2,36
170 POKE V3+1,18: POKE V3+2,170

This sets the 'envelope' for each voice. Voice 1 is bell-like; voice 2 sounds something like a harmonica; and voice 3 has the sound of a bass guitar.

As part of our POKE sequence we set the following characteristics:

- **attack** – how fast the sound appears. This will be fast for the bell and guitar, rather slow for the harmonica.
- **decay** – how quickly the initial sound settles. This will be fairly slow for the bell, fast for harmonica and guitar.
- **sustain** – at what level the sound will 'hold' once it's settled in. Bell and guitar won't hold the sound at all; it must fade away. The harmonica can keep the sound going for some time.
- **release** – how long it will take for the sound to die away. The bell dies almost instantly; the harmonica holds for a brief period; and the bass guitar continues to sound longer after it has been plucked.

180 T=TI

```
300 DATA 40,"  HAP",34,75,0,0,0,0
310 DATA 20,"PY",34,75,0,0,0,0
320 DATA 60," BIRTH",38,126,28,214,5,185
330 DATA 60,"DAY",34,75,28,214,0,0
340 DATA 60," TO",45,198,38,126,5,185
350 DATA 60," YOU",43,52,30,141,4,73
360 DATA 60,",",0,0,0,0,0,0
370 DATA 40,"  HAP",34,75,0,0,0,0
380 DATA 20,"PY",34,75,0,0,0,0
390 DATA 60," BIRTH",38,126,30,141,6,108
400 DATA 60,"DAY",34,75,30,141,0,0
410 DATA 60," TO",51,97,34,75,4,73
420 DATA 60," YOU",45,198,28,214,5,185
430 DATA 60,",",0,0,0,0,0,0
440 DATA 40,"  HAP",34,75,0,0,0,0
450 DATA 20,"PY",34,75,0,0,0,0
460 DATA 60," BIRTH",68,149,22,227,5,185
470 DATA 60,"DAY",57,172,25,177,0,0
480 DATA 60," DEAR",45,198,28,214,7,53
500 DATA 60," AN",21,154,30,141,7,163
510 DATA 60,"DREW",19,63,30,141,0,0
520 DATA 60,",",0,0,0,0,0,0
530 DATA 40,"  HAP",61,126,0,0,0,0
540 DATA 20,"PY",61,126,0,0,0,0
550 DATA 60," BIRTH",57,172,34,75,8,147
560 DATA 60,"DAY",45,198,28,214,0,0
600 DATA 60," TO",51,97,30,141,4,73
610 DATA 60," YOU.",45,198,28,214,2,220
620 DATA 0
1000 PRINT S$;:IF RIGHT$(S$,1)<"0" THEN PRINT
1010 RETURN
```

Happy Birthday









We'll be using the timer TI to control the timing of each note. Synchronize your watches; here comes the music-playing part.

```
200 POKE V1,16: POKE
    V2,32 : POKE V3,16
```

Release all the instruments – let go of the string, stop blowing, take your finger off. the keys. We must do this each time before we can start the next note. You might observe that the bell and guitar use a value of 16 for a mellow 'triangle' sound; the harmonica uses a more raspy 'sawtooth' created by value 32.

```
210 READ S: IF S=0
    GOTO 290
212 READ S$: GOSUB
    1000
```

Read the timing of the next note. If it's zero, quit. Read the word or part of a word that goes with a note. Print it (this time in white).

```
220 READ X1, Y1, X2, Y2,
    X3, Y3
```

Get the pitch of each of the three notes. If an instrument is silent this time around, its pitch will be given as zero in the DATA statement.

```
230 IF X1 THEN POKE
    H1, X1: POKE L1, Y1:
    POKE V1, 17
240 IF X2 THEN POKE
    H2, X2: POKE L2, Y2:
    POKE V2, 33
250 IF X3 THEN POKE
    H3, X3: POKE L3, Y3:
    POKE V3, 17
```

For each instrument: if its pitch is not zero, set the pitch and 'hit' the note. You will see that we make the note sound by adding 1 to the waveform value – compare these values with the ones shown in line 200.

```
260 T=T+S
270 IF T>TI GOTO 270
280 GOTO 200
```

We calculate the note's timing, and wait until the proper amount of time has passed. Then we go back and get the next note.

```
290 FOR J=L1 TO 54296:
    POKE J,0: NEXT J
295 PRINT CHR$(154):
    END
```

Finally, we clear all the SID music registers, change the printing colour back to light blue, and stop.

Here comes the DATA statements to play the music and write the words. Note that whenever a word ends with a full stop or comma, it will be printed and then a new line will be started. Finally, we see a subroutine at 1000 to print the word or part word, and to test if it ends in a non-alphabetic character. If so, a new line will be started. Be sure to include the semicolon after the PRINT statement in line 1000.

ANIROG SOFTWARE

PAYMENT BY CHEQUE, P.O., ACCESS/VISA
ANIROG SOFTWARE
26 BALCOMBE GARDENS
HORLEY, SURREY
ENQUIRIES: HORLEY (02934) 2007/6083

TRADE ENQUIRIES
29 WEST HILL, DARTFORD, KENT. (0322) 92518

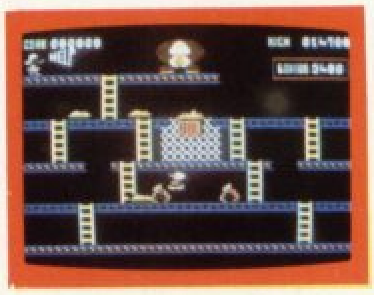
COMMODORE 64

KONG KB / JS £7.95

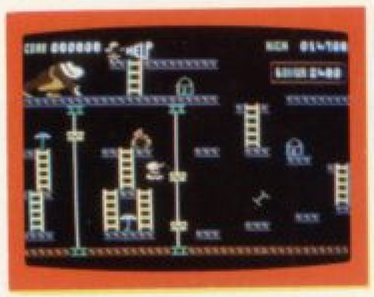
A 31 K ALL M/C PROGRAM WITH BRILLIANT ANIMATED GRAPHICS



1. CHASING FIREBALLS AND BARRELS, CAN BE SMASHED BY THE MALLET IF YOU CAN JUMP UP AND GET IT.



2. JUMP OVER THE PIES AS YOU RUN ALONG THE CONVEYOR BELT MAKE SURE YOU DONT END UP INSIDE THE COOKER.



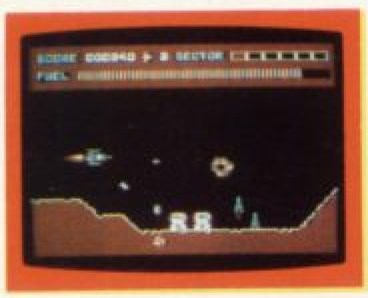
3. THE ONLY WAY UP IS BY THE LIFT WHICH IS UNDER A CONSTANT BARRAGE OF GIRDERS.



4. KNOCK OUT ALL THE SUPPORTS BY RUNNING OVER THEM AND WITNESS THE SPECTACULAR COLLAPSE OF KONGS LAIR.

SKRAMBLE J.S. £7.95

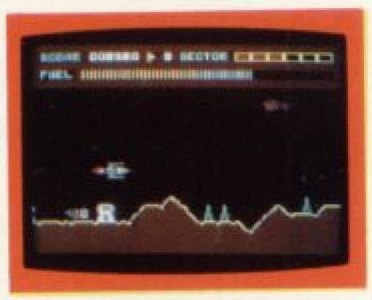
PILOT YOUR SPACE CRAFT THROUGH THE SIX DEFENCE SECTORS AND DESTROY THE COMMAND MODULE OF THE **COBRON** EMPIRE. A 32 K ALL M/C BLOCKBUSTER.



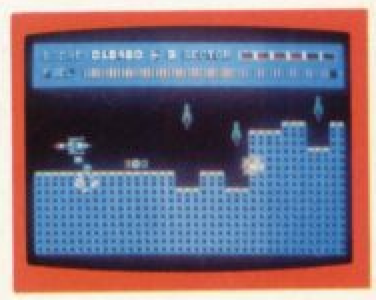
1. RAMMING MISSILES HURTLE TOWARDS YOU.



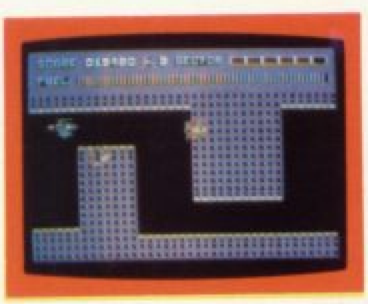
2. FIGHT COBRON QUASER ION SHIPS.



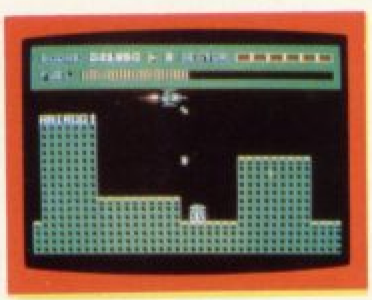
3. DODGE THE FIREBALLS, BECAUSE YOU CAN'T SHOOT THEM DOWN.



4. CITY WITH HIGH RISE BUILDINGS.



5. GUIDE YOUR SHIP WITH DEXTERITY THROUGH THE MAZE.



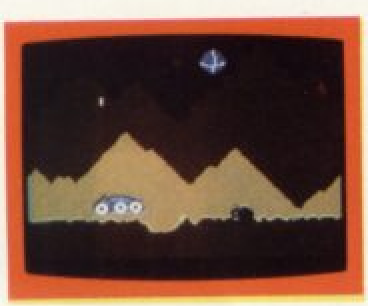
6. DESTROY THE BASE BEFORE YOU RUN OUT OF FUEL.

HEXPERT J.S. £7.95

THE EVIL WITCH ZOGANAAR HAS FORCED BERT TO PLAY ON HER 3D HEXAGONAL PYRAMID. AS HE JUMPS FROM STEP TO STEP OF THE PYRAMID. MASSIVE RUBBER BALLS BOUNCE DOWN THE PYRAMID TO CRUSH HIM. HER PET SNAKE, **COILY**, MOVES UP AND DOWN THE PYRAMID CHASING BERT. YOU HAVE TWO SPINNING DISC'S TO JUMP-ON TO AND LURE **COILY** TO HIS DEATH. CAN YOU MAKE BERT THE HEXPERT?
A SUPERB ALL M/C GAME WITH EXCELLENT 3D GRAPHICS.



MOON BUGGY



EXPERIENCE ALL THE THRILLS OF THE ARCADE GAME AS YOUR PATROL CRAFT MANOEUVRES OVER THE GIANT POT HOLES.

J.S. £7.95

3D TIME TREK



A 3D STAR TREK GAME WITH REAL ARCADE ACTION BATTLE WITH THE ALIENS.

K.B. / J.S. £5.95

DUNGEONS



ENTER THE REALMS OF FANTASY WITH SUPERB USE OF SPRITE GRAPHICS BATTLE WITH THE FEARFUL FIRE BREATHING RED DRAGON

K.B. £6.95

FROG RUN



A POPULAR ARCADE GAME BROUGHT TO LIFE WITH BRILLIANT ANIMATED GRAPHICS

K.B. / J.S. £5.95

Coming soon

The winter always sees a lot of shows and exhibitions, and the kind of people who organise these things are belatedly latching on to the fact that home computers are A Big Thing. Here's a round-up of those we've come across – some new shows plus the old familiar names.



28 Sept-2 Oct London

Personal Computer World Show. The big one – potentially uncomfortable mix of home computer stuff and more business-oriented exhibits, but somehow it works. Various beat-the-wait deals on tickets; otherwise you'll probably have to queue. Organisers: Montbuild for PCW, 01-486 1951.

6-8 November Glasgow

Scottish Home Computer & Electronics Show. Another new one – includes CB, hi-fi, video, ham radio as well as computers. In the Anderston Centre. Organisers: Trade Exhibitions Scotland, 0764 4204.

11-13 November Bristol

HomeTech 83. Again a new one, held in the rather nice Bristol Exhibition Centre and promising exhibitors with "computers, software, books, video games or other accessories". Organisers: Tomorrow's World Exhibitions, 0272 292156.

24-26 November Manchester

Northern Computer Fair. Mix of business and home computing. Organisers: Reed Exhibitions, 01-643 8040.

15-18 December Wembley

'Your Computer' Christmas Fair. Another first-timer, another magazine spin-off. Aims to be "Britain's major pre-Christmas showcase of home computing". Organisers: Reed Exhibitions for Your Computer, 01-643 8040.

9-13 November Dublin

Home Computer Show. And another new one – at the Royal Dublin Society. Organisers: SDL Exhibitions, Dublin 763871.



CROSSWORD ENTRY FORM

Name: ....

Address: ....

....

....

Size: ☐S ☐M ☐L ☐XL

Send to: Crossword Competition, Commodore User, The Metropolitan, Enfield Road, London N1 5AZ.

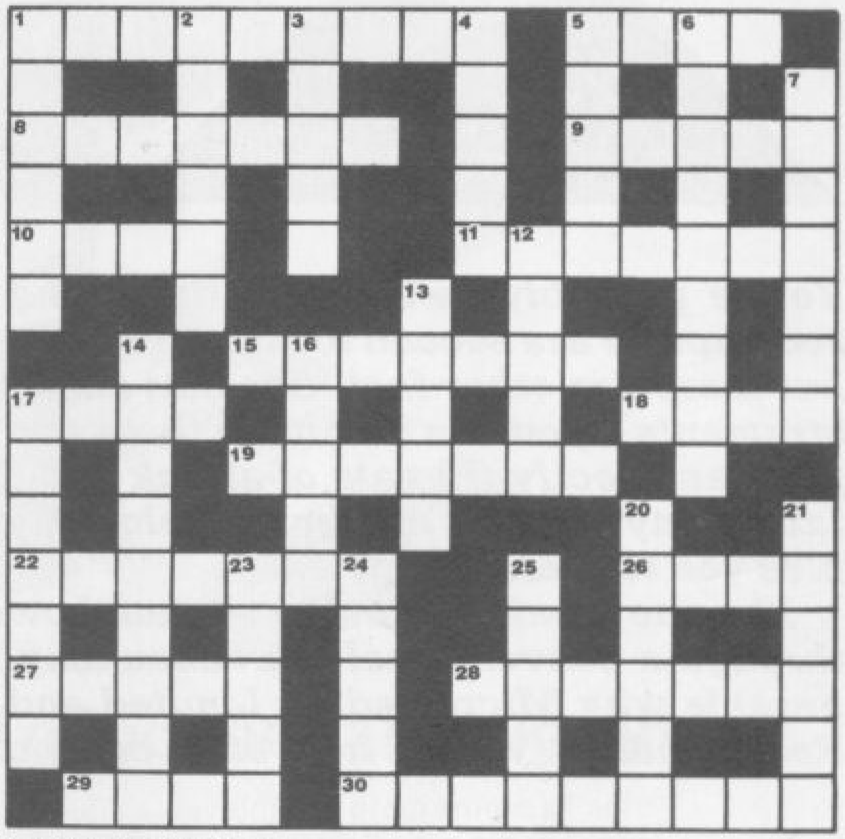
CROSSWORD

*Welcome to **Commodore User's** exciting new regular competition. We kick off with a tantalising crossword to fill in during those rare moments when you're away from the computer.*

We're giving you the chance to win yourself a smart, new (well, freshly laundered) Vicwear sweatshirt!!!

Prizes go to the first five correct entries to arrive by 14th October to: Commodore User, The Metropolitan, Enfield Road, London N1 5AZ. (Please mark your envelope Crossword Competition.*) All completed crosswords must be accompanied by an entry form – (remember to state your size, S, M, L, XL)*

Answers, winners and a new contest in next month's issue. Meanwhile, happy solving. Friends and accomplices of Paradox are excluded from all competitions.



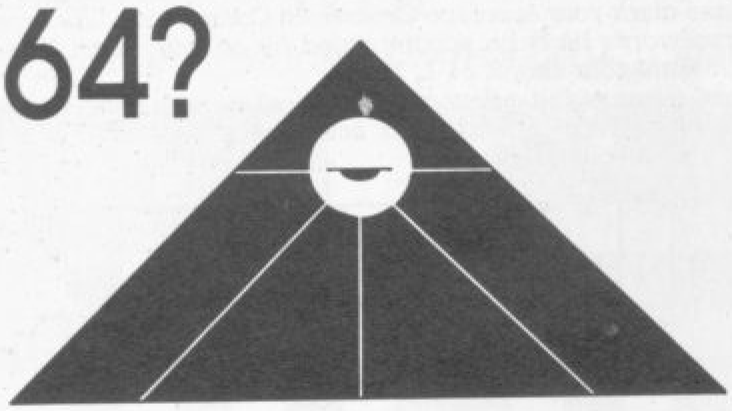
CLUES ACROSS

1 Vic rival – flee! (9)
5 Initial method of making oven dishes edible in operating style (4)
8 Multi-user fishing? (7)
9 In turn as a logic, a bit nosey sounding (5)
10 Almost Irish, this flag. (4)
11 Not very arousing – could result in a cold start! (4,3)
15 Sticks he reads about. (7)
17 Change of program flow results in a nervous twitch. (4)
18 Register ending after the French look. (4)
19 Arrange part of a disk to affect a certain character. (7)
22 Lessens secured agitations. (7)
26 Interface that is bidirectional. (4)
27 Confused Atari and kept on top. (5)
28 Home of a byte maybe. (7)
29 Network in which I remain silent. (4)
30 & 12 All Vic and 64 owners should be reading them. (9,5)

CLUES DOWN

1 & 17 Drain sent Jet round the editor. (6,7)
2 Travel to Silicon Valley and get lost... (2,4)
3 ...then turn through 90 degrees to go here. (5)
4 Yes, at first a computerised high-technology sailing vessel. (5)
5 Logical operator after a short time is subordinate. (5)
6 Plunges around rising loss and goes into solution. (9)
7 Learner in temporary storage – he'll pull the wool over your eyes. (7)
12 See 30 across.
13 Records after he assists. (5)
14 Operate without a delay! (9)
16 Saver with a lot of energy. (5)
17 See 1 down.
20 With German communist, joint is at an angle. (6)
21 Minus two points if you're not the owner. (6)
23 Program accident. (5)
24 Relative with integrated circuit might be super! (5)
25 Army's masters use measure in telecommunications. (5)

Even better sound from the 64?



The Hi-Res Soundbox tested

You're probably aware that the sound characteristics of the Commodore 64 microcomputer are second to none. You can set the volume level and the pitch of the note, but there are other facilities that enable you to simulate many different musical instruments – you can nominate the wave form from smooth (triangle) to harsh (pulse): you can specify the rate of attack and decay, the speed the note reaches and how it falls away from its maximum volume; you can declare a sustain setting, which enables you to prolong a note...

The one weakness in the system however, is that those superb sounds are played through a conventional television amplifier and its puny loudspeaker. This (it says here) is why Microtrading Limited and Potter Sound jointly have developed the Hi-Res Soundbox. We got hold of an early unit for review.

In essence the Hi-Res Soundbox simply bypasses the TV amp and speaker to produce the 64's sound output with much improved quality.

The thing is a box containing one of the new-generation mini-loudspeakers with a volume control on it and a connection to the Vic. There's also a separate power lead (yet another socket is needed to run the computer!); inside is a small but impressive amplifier. The casing is plastic but pretty solid. It is finished in battleship grey, which is functional rather than elegant – something like matt black or even a brown to match the 64 would have been better, but maybe future versions will have a different finish.

For the technically minded, the unit incorporates a high-frequency tweeter and a four-inch base woofer; these will accurately reproduce a full range of sounds from 50Hz to 10 Kilohertz, and the difference in quality of sound when compared with the office's portable TVs is marked.

Output is 10 watts RMS with a 20 watt peak, and 10 watts must be quite loud enough for anything you're doing on the 64 – the sound reproduction at peak volume is very good, with little or no distortion.

Inside the layout and assembly look neat, always a good sign for reliability. The Hi-Res Soundbox is simple to fit, of course, and requires no extra programming. Ours came with a demo tape that sounded good, and running our own music through it gave enough clarity to force a few rewrites to improve the orchestrations. The Hi-Res Soundbox was particularly impressive with voice output, too.

So far so good. On the minus side, our unit stopped working almost as soon as we plugged it in; there was a faulty connection to the volume control knob, but we fixed that. We're told that production units have a slightly different and more robust internal design.

The only other cavil is the price, nearly £50. We don't begrudge that too much for the unit itself, even though at that kind of money you should be able to get a *pair* of similar loudspeakers for your stereo: after all the Hi-Res Soundbox does include an amp, and it does offer instant plug-in-and-go connection to the 64. No, our only thought is that you could very easily do it yourself by putting the audio output to



a stereo system... *if* you have one available, and *if* you're halfway handy with a soldering iron and the AUX connection on your stereo amp.

EASY SCRIPT REVIEW



by Nigel Ruth

Easy Script is Commodore's own word processor for the CBM 64. At present it is only available on a disk; the manual has been written for a cartridge version, however, which would facilitate the use of both cassettes or disks as the storage medium for the text.

There is sufficient memory to store about six full pages of text in memory at any one time. This is not an absolute limitation, though, as files of text can be saved and linked automatically on printing to give an unlimited length of output.

Easy Script operates in three modes – *edit*, *disk* and *command*. Edit mode is the one used to input and edit text: disk mode is for loading and saving files, viewing directories, and performing other disk operations such as file deletion. Command mode is used to instruct Easy Script to perform various functions like printing the text, inserting lines or setting tabs.

At the top of the screen the current status of the program is displayed, including the mode in which the program is running and a range of error messages. When the user is in edit mode the current line and column number of the cursor is displayed.

Using the program

After typing LOAD "EASY SCRIPT", 8, 1 and hitting the Return key the whole screen fills with the message LOADING EASY SCRIPT in large block capitals. This stays on for a short delay, then the screen clears and you are presented with some self-explanatory input prompts.

The first question asks you to specify the number of columns of text on the screen display. Although the screen is 40 columns wide, Easy Script allows you to set widths from 40 to 240 columns. This is particularly useful when you are preparing tables. This question has no effect on the output produced by the program on the printer – full use can be made of the 80 columns available there.

The next prompt asks whether you are using tape or disk as the media for storing the text.

And the final question asks what type of printer you are using – you're given a choice of five:

0 CBM
1 MX80
2 SPINWRITER
3 QUME/DIABLO/8300
4 Other

After all these questions have been answered to set up the system you are entered automatically into edit mode. It is now possible to adjust the colour combination of the screen, the border and the cursor to whatever you find pleasant and comfortable for viewing.

Text creation on Easy Script is identical to that of the computer when it is first turned on – cursor up, down, right, left and the delete key all work as normal. As you enter text a flashing cursor follows you around the screen.

Typing the text with Easy Script is easier than using a typewriter in that you enter the text continuously, pressing the Return key only at the end of a paragraph. When the Return *is* pressed a reverse-field back arrow is displayed at that place in the text and the cursor automatically jumps to the beginning of the next line.

Functions

Search and replace is an interesting feature that is (or should be) standard on all WP packages worth the name – on Easy Script it allows you to replace all the occurrences of a word or sentence of maximum length 32 char-

Easy Script Commands

MODES	
EDIT On/Off	F1
Sound On/Off	F1/*
DISK	F4
Display directory	$0 or $1
Load directory	+$0 or 2$1
Delete file (drive X)	sX:filename
Restart program	F1/RUN/STOP
CURSOR CONTROLS	
Right	CRSR →
Left	SHIFTed/CRSR←
Down	CRSR↓
Up	SHIFTed/CRSR↑
Up a line	←
Start of line	←/CRSR→
DISPLAY	
Next Screen	F1/space bar
Previous screen	F1/SHIFTed space bar
GO TO	
Start of file	SHIFT/CLR/HOME
Home	CLR/HOME
End of text	F1/G/E
Line xxx	F1/G/xxx
Max. line number	F1/G/999
PANNING	
Up	F1/SHIFTed CRSR ↑
Down	F1/CRSR ↓
Left	F1/SHIFTed/CRSR←
Right	F1/CRSR →
stop	RUN/STOP
Speed up	SHIFT
Hold/Restart	tap space bar
TEXT	
Delete:	
characters	INST/DEL
lines	F1/INST/DEL (f1 off)
block	F1/D/CRSR ↑ (R)
Erase:	
character	space
sentence	F1/E/S
paragraph	F1/E/P
remainder	F1/E/R
all	F1/E/A
Insert:	
space	SHIFT/INST/DEL
blank lines	F1/SHIFT/INST/DEL (F1 off)
mode	F1///I (F1/I off)
Range:	
mark block	F1/R/CRSR
memorize	(R)
transfer	F1/X
repeat (again)	F1/A
save block	F1/SHIFT/F/ blockname
Search:	
specify	F1/S
replace	F1/@/M or L
hunt	F1/H/M or L
Soft hyphen	F1/-
Linking space	SHIFTed space
Horizontal Tabs:	
set	position cursor F1/T/H
go to tab	F7
clear	position cursor F1/C/H
zero all	F1/Z/H
Vertical tabs:	
set	position cursor F1/T/V
go to tab	F8
clear	position cursor F1/C/V
zero all	F1/Z/V
Decimal tabs (set)	position cursor F6/F1/T/H/F6
Display tabs	F1/P
FILES	
Load	F1/L/filename
Save	F1/F/filename
Merge	F1/I/F1/L/filename
FILL FILES	
Variable block	F1/B
Measured block	F1/B/cursor back/ F1/M
Fill blocks from cursor	F1/V
Empty blocks	F1/SHIFTed V
Go to next block	F1/F7
Close fill file	F1/CLR/HOME
Print, blocks filled	F1/O/F/C/P/fill filename(R)
OUTPUT	
To video or printer	F1/O/V or P
Continous	F1/O/C/V or P
With fill file	F1/O/F/V or P
With linked files	F1/O/L/V or P
Multiple copies	F1/O/X/#(R)/P
Stop	RUN/STOP
At end of page:	
Next page	C
Change, video/ print	V or P
Change to continuous	SHIFTed P
PRINT FORMAT	
Underline or enhance	F1/((F1/ off)
Reverse field	F1/((F1/) off)
Pause during print	ps

acters with another word or sentence of up to 32 characters.

This can also be done within a set of linked files. It is very useful if you have made a spelling mistake throughout a manual – you enter the word as you have wrongly spelt it and then the word as it should be spelt and the error will be quickly corrected.

The program will even recognise text which has been split over two lines, which is by no means the case with all word processors – including some of the more expensive and well-known best-sellers.

To allow you to rearrange blocks of text within your document there are four simple commands. There is even a command to allow you to save a particular block while ignoring the rest of the document.

If an entire word does not fit at the end of a printed line, a 'soft hyphen' can be used to show where it should be split. This will obviously be ignored if the word does not need to be split.

The function keys form an important part of the program's operation: they are used for instance to change the mode of the program, to turn the sound on and off, and to turn the capitals on and off.

Unfortunately the program does not allow the user to make use of the extensive graphics available directly from the keyboard. Instead each time you want a graphic character you have to define it from scratch yourself.

Formatting

A major contribution to the success and popularity of word processors lies in the fact tht you can extensively format a document using simple 'in-text' commands which can be placed anywhere.

All of these formatting commands must be preceded by a reverse field asterisk (obtained by hitting F3). It has become standard practice when I am typing in a document to have a list of these commands at the start of the document setting up such parameters as the page length of the output, the position of the left and right margin, and (usually) turning the justification on.

Most of these two-character letter mnemonics are easily remembered. You can see that from this list of some of these formatting commands:

Justify the text	jul (ju0—off)
Centre the text	cnl (cn0—off)
Align the text to the right	ral (ra0—off)
Set width of left margin	lmxx
Set width of right margin	rmxx
Release the margin	maxx
Set text length	tlxx
Set linc spacing	spxx
Vertical positioning	vpxx (vo0—off)
Form feed to the end of the page	fp0
Set the typing pitch	ptxx
Set the number of lines per inch	lpxx
Define heading	hdxx:text,,
Define footing	ftxx:text,,
Set the page number	pxx
Set the number of lines inch	lpxx

Storage

All the text created by Easy Script is stored as sequential files either on disk or cassette and can be easily accessed from the program. Directories can be looked at, disks formatted and files erased without disturbing any text in the memory. Files can also be merged by the use of a simple command.

Crashproof

I have been experimenting with the program and its many commands for quite a considerable time now: and so far Easy Script appears to be totally crashproof. And when I did accidentally make mistakes, it was able to tell me exactly what I had done wrong (like not switched on the printer or disk drive).

Printing

There are quite a few interesting commands available to control the printer, including 'pause' – this stops the printer during printing to allow you for example to change the ribbon to a different colour: when 'C' is pressed on the keyboard the printing will proceed.

Multiple copies can easily be obtained by specifying the number required. A standard letter can be merged with another file containing people's names and addresses to obtain personalised letters.

All the facilities available on the printer are made extensive use of including the enhancement of characters but excluding the standard graphics available normally.

Conclusions

From what you have just read it should be fairly clear that I think Easy Script is a very good word processor. The price tag of £75 might be slightly expensive for an occasional user at home: but it would be excellent value for money in an office which gets through a larger quantity of typing.

I found Easy Script laden with facilities, simple to learn and provided with documentation that is very comprehensive. I would urge anyone considering purchasing a word processor to look very seriously at Easy Script.

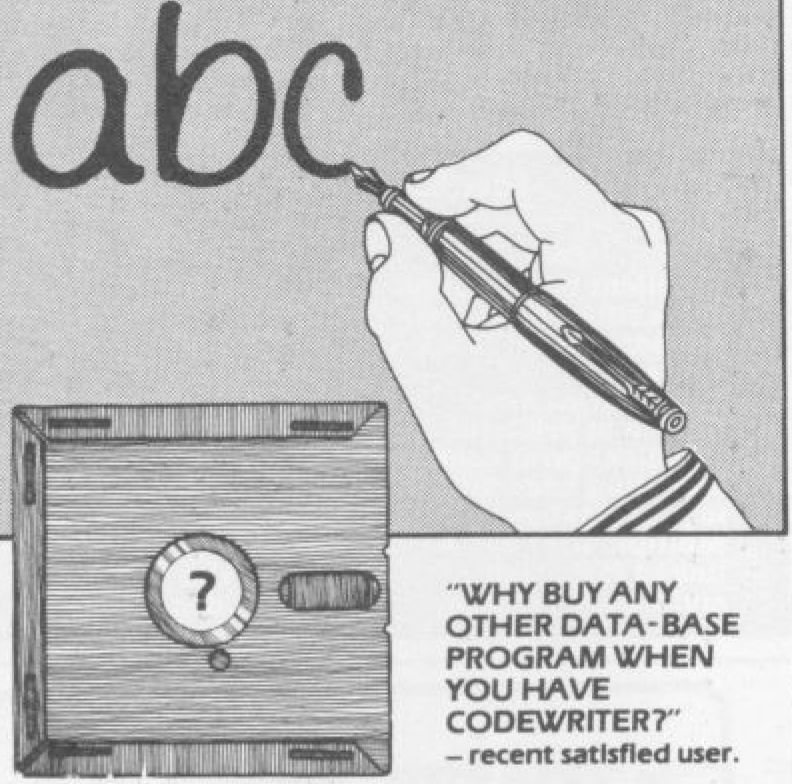
Easy Script on balance

For
- Excellent text creation and editing functions, including good block operations.
- Excellent manual.
- Choice of printers.
- Excellent print controls including headers, titling, pagination.
- Multiple files can be linked for a single multi-page print run.
- Can be used in conjunction with Easy Spell to check for spelling mistakes.

Against
- Price.
- Display becomes cluttered with non-text format controls.
- Does not allow the user to make use of the 64's graphics.

CodeWriter

makes data-base and related programming as simple as



"WHY BUY ANY OTHER DATA-BASE PROGRAM WHEN YOU HAVE CODEWRITER?"
– recent satisfied user.

Forget about the expense of hiring a professional programmer or the restrictions of 'off-the-shelf' programs. Become your own program designer producing microsoftware to suit your specific requirements. Codewriter enables you to do this, you type commands in your own everyday language, just as you would write them on paper . . . it's as simple as ABC!

When you've completed your design, Codewriter will write the program code and store it on your own disk whilst you sit back and watch. At anytime, you can of course modify the programs you've designed.

Your programs may include data bases printed reports, calculations and comparisons between fields of data, development of menus, forms, letters, memos, cheques, invoices, statements, mailing labels . . . the possibilities for design are endless. Codewriter operates with most popular micros and will be demonstrated fully at your local dealer.

For details on Codewriter, complete and return the coupon to;

DYNATECH MICROSOFTWARE LTD.

Summerfield House, Vale, Guernsey,
Channel Islands.
Telephone 0481 45934 Telex 4191130.

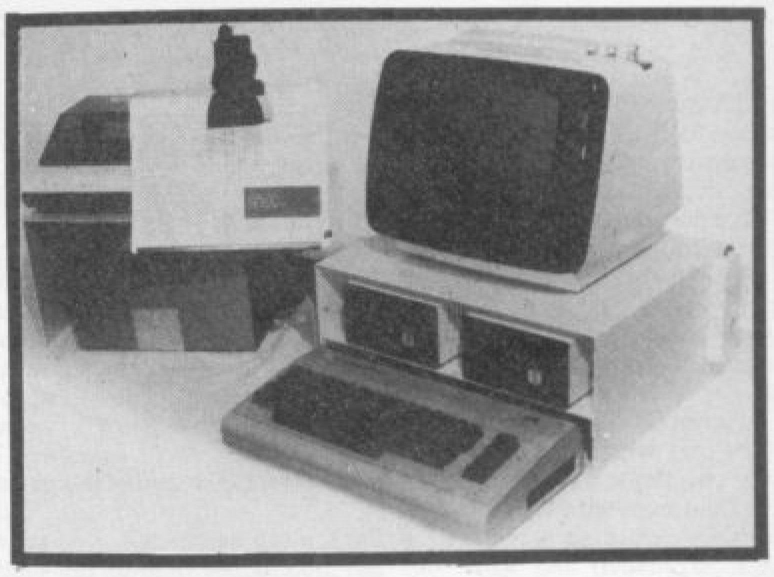
name ____________________

address ____________________

____

____________ tel. ____________ C.U.

NEWSCOPE DEVELOPMENTS LTD



A—STACK

TV/MONITOR STAND which includes a shelf for disc drives etc and has space to stow a micro when not in use. Strong steel construction with attractive matt cream finish. Size 430mm wide, 200mm high, choice of 350mm, 300mm, or 250mm deep.

Price: £22.00 **Postage & Packing: £4.00**

PRINTER STAND

Allows printer to stand over a 2000 sheet box of continuous listing paper. Same colour and finish as A—Stack, also made of steel.
Dimensions are height = 210mm high, depth = 250mm, width = 430mm.

Price: £14.50 **Postage and Packing: £3.00**

BOOK REST

Lightweight rest for books, listings etc fitted on vertical hinge which can be attached to the left or right side of the A—Stack. Same colour and finish as above.

Price: £7.99

MULTI-WAY POWER BLOCK

Up to four separate power supplies for computer and peripherals from a single mains power point.
Can be fitted to A—Stack and is supplied with four connectors and 2 metre mains cable.
Only 150mm x 45mm.

Price: £11.99

CLIP-ON LAMP

Lightweight mini-spot lamp which can be fitted as required to illuminate book rest and for computer keyboard.
Available in brown or black.

Price £8.99

If all five items ordered: Postage and Packing FREE SAVE £7.00

Orders to:-
Victory House, 8a The Rank,
North Bradley, Trowbridge, Wilts,
Tel: 02214 4425/0373 864644

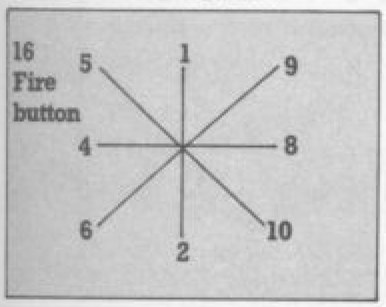
Controlling a Joystick the easy way

STick is a short machine-code program that will aid any Vic user who wants a simple method of reading the position of the joystick from Basic. It runs in any memory configuration.

When RUN, it creates 64 bytes of machine-code starting at location 673 decimal ($02A1 hex): subsequently calling this via **SYS 673** activates the program. From this point onwards the machine code is running, and can only be stopped by turning the Vic off or by hitting the RUN/STOP and RESTORE (SYS 673 to re-engage it).

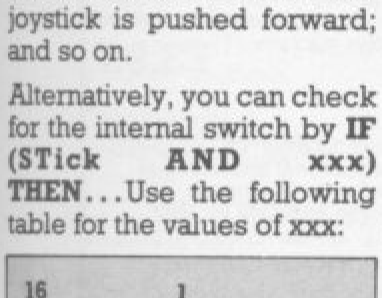
The result is that the (rarely used) system variable **ST** (status) is converted into a more useful command called STick (this is possible because only the first two letters of a variable name are significant to the Vic). Now the joystick can be read by an **IF STICK=xxx THEN** statement.

The values of STick can be read off this diagram:



So if STick=1 then the joystick is pushed forward; and so on.

Alternatively, you can check for the internal switch by **IF (STick AND xxx) THEN**...Use the following table for the values of xxx:



By ANDing STick with these values, the internal joystick switches can be checked for.

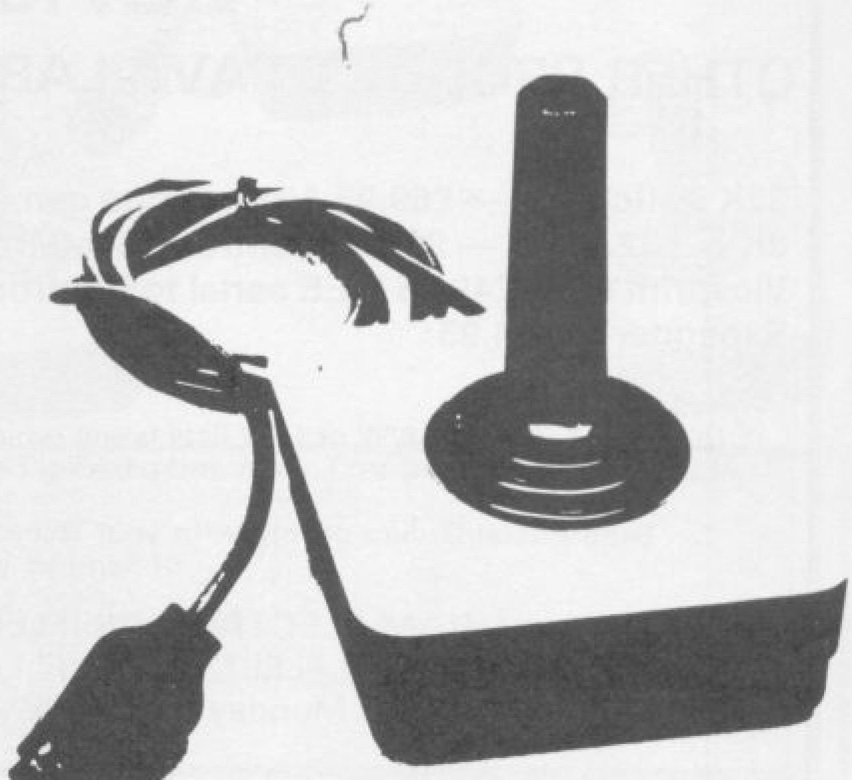
Example Program

```
10 SYS 673: REM engage
   STick
20 IF (STICK AND 1)
   THEN PRINT "UP"
30 IF (STICK AND 2)
   THEN PRINT "DOWN"
40 IF (STICK AND 4)
   THEN PRINT "LEFT"
50 IF (STICK AND 8)
   THEN PRINT "RIGHT"
60 IF (STICK AND 16)
   THEN PRINT "FIRE!"
70 PRINT: FOR I=0 TO
   200: NEXT: REM delay
80 GOTO 20
```

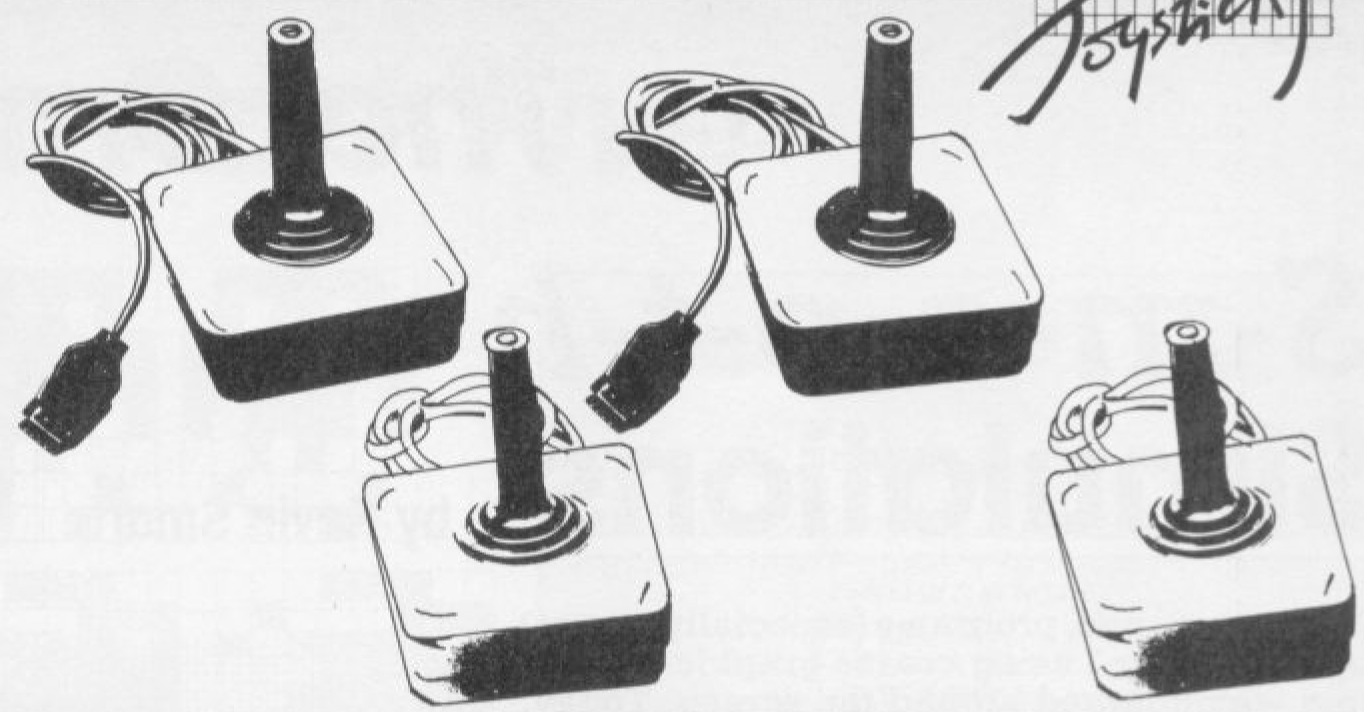
In the Basic Loader program, the variable **S** gives the start of the machine code. This has been set at 673 as normally this area of ROM is totally unused by Basic; but by changing the value of S the code can be relocated anywhere in RAM. The program will work on a Vic with any memory size — but problems may be encountered if using a ROM cartridge simultaneously.

Program details

10-80	**DATA for the machine code**
100	**sets T to 0 and sets starting location of code**
120	**gets one item of data**
130	**performs the calculations to relocate the code**
140	**POKEs in machine code and keeps a running score of the data read in T**
150	**checks T to see if the data has been entered correctly**
160	**engages STick**







How the machine code works

The initial SYS command sets up a new interrupt vector, as controlled by locations 788 and 789 ($0314 and $0315). This points to the start of the joystick interpreter program (resident at S+13 bytes).

Every time one IRQ interrupt is created thereafter, the joystick program is called. This reads the position of the stick from the VIA chips, juggles the numbers conveniently together, and stores the result in memory location 144 ($90) — which contains the value of the STatus variable.

After this the program jumps to the start of the conventional interrupt program at 60095 ($EABF). The interrupt rate is set by VIA No 2, and is normally about 60Hz — which means that the position of the joystick is checked 60 times a second, even while using Basic!

The only disadvantage of this routine is that the STatus variable cannot now be used. But as this comand variable rarely crops up in home programming, it's loss should not be too great.

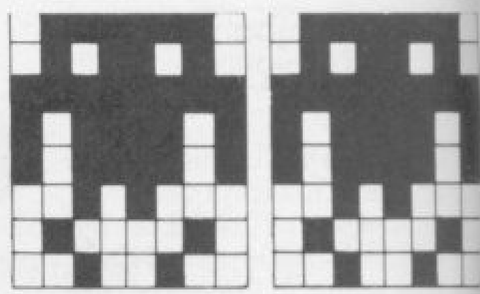
```
10 DATA 120,169,0,141,20,3,169,***
20 DATA 141,21,3,88,96,169,0,141
30 DATA 19,145,141,34,145,173,17,145
40 DATA 41,31,74,74,133,144,173,17
50 DATA 145,41,32,74,5,144,133,144
60 DATA 173,32,145,41,128,74,74,74
70 DATA 74,5,144,73,31,133,144,169
80 DATA 255,141,34,145,76,191,234,234
90 PRINT"[CLR] LOADING MACHINE CODE"
100 T=0:S=673
110 FORI=0TO63
120 READA$
130 IFA$="***"THENPOKES+I,(S+13)/256:POKES+I-5,(S+13)AND255:NEXT
140 POKEI+S,VAL(A$):T=T+VAL(A$):NEXT
150 IFT<>6329THENPRINT"[CUD][CUD]TYPING ERROR IN DATA.":END
160 SYSS
```

Review

The Graphics

S-m-o-o-t-h Simulations

by Kevin Smart





A few years ago, programs (especially games) were produced using coarse graphic characters which jerked around the screen. Today, we have lifelike user-defined characters which *also* jerk around the screen!

Some programs do have an improved stationary graphics quality: but when things begin to move, the accuracy gained through visual appearance is virtually lost. (There are some exceptions to this, notably in professional software such as Commodore's own cartridges.) I would like to see this standard in all games/programs and I hope that these articles are improving things.

In most cases, the idea for one of these articles comes through seeing professional software. I try to produce the same effect and once I'm satisfied, I write this. The program which follows was produced after seeing VIC AVENGERS: I wondered how the smooth animation was achieved.

At first I thought that a separate character was defined for each slight movement of the invader in VIC AVENGERS. A program was written and it did indeed work. On inspecting the situation, though, I concluded that this method was not economic if I wanted to use such a capability in games — because 16 characters would have to be defined just to move an object from one character space to another.

A close look at the *Programmer's Reference Guide* revealed that there were machine-code instructions suited to my needs — you can shift the bits in each of the eight bytes of the defined character in a left or right direction. These are the commands ASL, ROL, LSR and ROR.

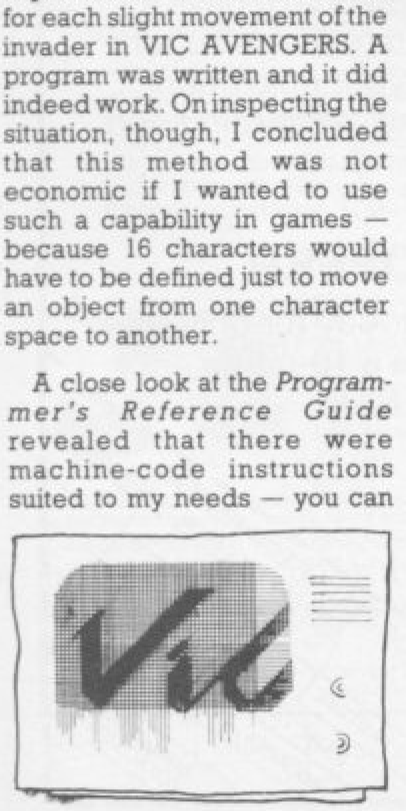
I found the machine code fairly easy to write because of its small size. And for once it worked first time.

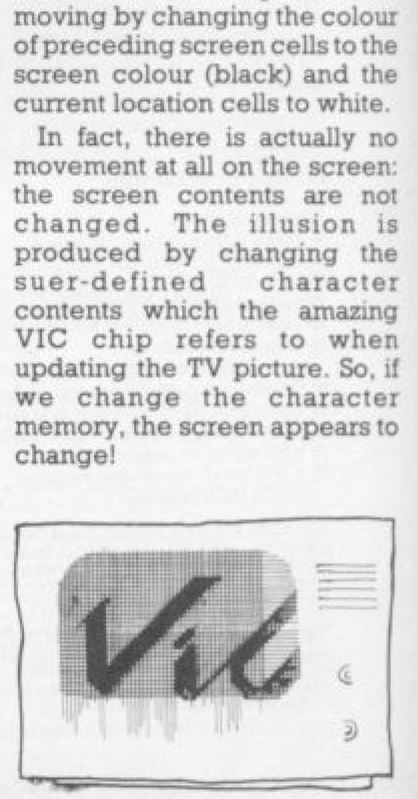
The program uses only two defined characters to produce the marvellous effect; it gives the illusion of a single invader moving by changing the colour of preceding screen cells to the screen colour (black) and the current location cells to white.

In fact, there is actually no movement at all on the screen: the screen contents are not changed. The illusion is produced by changing the suer-defined character contents which the amazing VIC chip refers to when updating the TV picture. So, if we change the character memory, the screen appears to change!

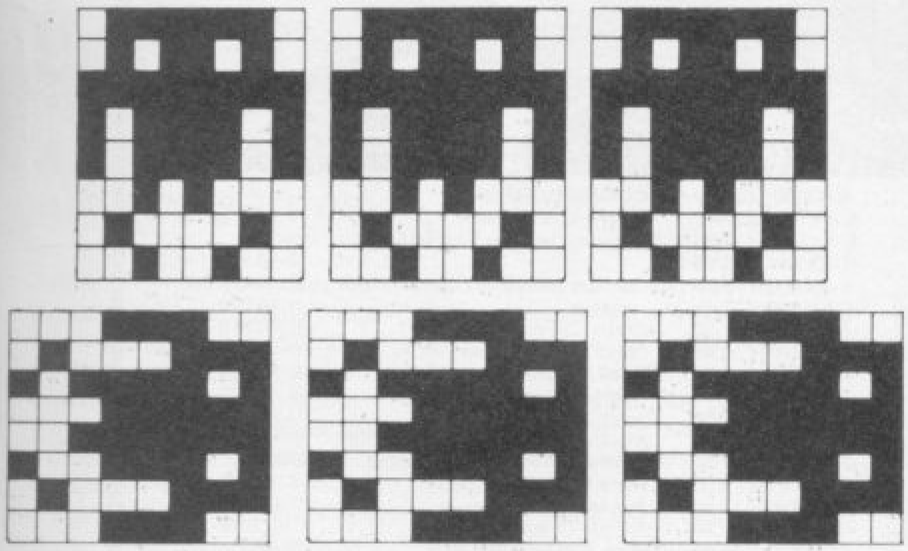
```
5 REM *** SMOOTH GRAPHICS ***
10 REM *** BY K.SMART 20.2.83. AGE 15 ***
20 REM *** RESERVE MEMORY  ***
30 POKE 56,28: POKE 55,0: CLR :CS=38400: S3=36876: S4=36877
40 FOR LOOP=7168 TO 7679: POKE LOOP,0: NEXT
50 PRINT"[CLR]": POKE 36879,12
60 REM *** DEFINE INVADER  ***
70 FOR LOOP=0 TO 7: READ CODE: POKE 7168+LOOP,CODE
80 NEXT LOOP
90 DATA  36, 126, 219, 126, 60, 36, 66, 36
100 REM *** READ MACHINE CODE ***
110 FOR LOOP=0 TO 83 : READ CODE: TEST=TEST+CODE: POKE 7200+LOOP,CODE
120 NEXT LOOP
130 IF TEST<>9446 THEN PRINT"[WHT]ERROR IN DATA. CHECK  DATA!": END
500 REM *** RIGHT SHIFT @ TO A CODE (7200)***
510 DATA 162, 8, 94, 255, 27, 126, 7, 28, 202, 208, 247, 96
520 REM *** RIGHT SHIFT A TO @ CODE (7212) ***
530 DATA 162, 8, 94, 7, 28, 126, 255, 27, 202, 208, 247, 96
540 REM *** LEFT SHIFT @ TO A (7224) ***
550 DATA 162, 8, 30, 255, 27, 62, 7, 28, 202, 208, 247, 96
560 REM *** LEFT SHIFT A TO @ (7236) ***
570 DATA 162, 8, 30, 7, 28, 62, 255, 27, 202, 208, 247, 96
580 REM *** DOWN (7248) ***
590 DATA 162, 15, 189, 255, 27, 157, 0, 28, 202, 208, 247, 169, 0, 141, 0, 28, 9
6
600 REM *** UP (7265) ***
610 DATA162, 0, 189, 1, 28, 157, 0, 28, 232, 224, 15, 208
620 DATA 245, 169, 0, 141, 15, 28, 96
800 REM *** DEMONSTRATION ***
810 PRINT CHR$(144)"[CLR]@A@A@A@A@A@A@A@A@A@A"; :PRINT"A@A@A@A@A@A@A@A@A@A@"

820 POKE36869,255: POKE36878,15
825 REM *** GO RIGHT ***
830 FOR B=0 TO 18 STEP2 :POKE CS+B,1: POKE CS+B+1,1: POKES4,127+B*7
840 FOR L=0 TO 7: SYS 7200: WAIT162,2: NEXT: POKE S3,220
850 POKE CS+B,0: POKE CS+B+2,1
860 FOR L=0 TO 7: SYS 7212: WAIT162,2: NEXT:POKE S3,130
870 POKE CS+B+1,0: NEXT: POKE CS+B,0
880 REM *** GO LEFT ***
890 FOR B=18 TO 0 STEP-2: POKE CS+B,1: POKE CS+B-1,1: POKES4,127+B*7
900 FOR L=0 TO 7: SYS 7224: NEXT
910 POKE CS+B,0: POKE CS+B-2,1
920 FOR L=0 TO 7: SYS 7236: NEXT
930 POKE CS+B-1,0: NEXT
940 REM *** DANCE? ***
950 FOR B=0 TO 16 STEP 4: POKE CS+B,1: POKE CS+B+1,1: POKES4,127+B*7
960 FOR A=0 TO 7: SYS 7200: WAIT162,8: NEXT: POKE CS+B,0: POKE CS+B+2,1: POKES3,
220
970 FOR A=0 TO 7: SYS 7212: WAIT162,8: NEXT: POKE CS+B+1,0: POKE CS+B+24,1: POKE
 S3,200
980 FOR A=0 TO 7: SYS 7248: WAIT162,8: NEXT: POKE CS+B+2,0: POKE CS+B+25,1: POKE
 S3,180
990 FOR A=0 TO 7: SYS 7212: WAIT162,8: NEXT: POKE CS+B+24,0: POKE CS+B+26,1: POK
E S3,160
1000 FOR A=0 TO 7: SYS 7200: WAIT162,8: NEXT: POKE CS+B+25,0: POKE CS+B+4,1: POK
E S3,140
1010 FOR A=0 TO 7: SYS 7265: WAIT162,8: NEXT: POKE CS+B+26,0: NEXT: GOTO810
```





s Primer Part Five



Right shift & to A(7200)

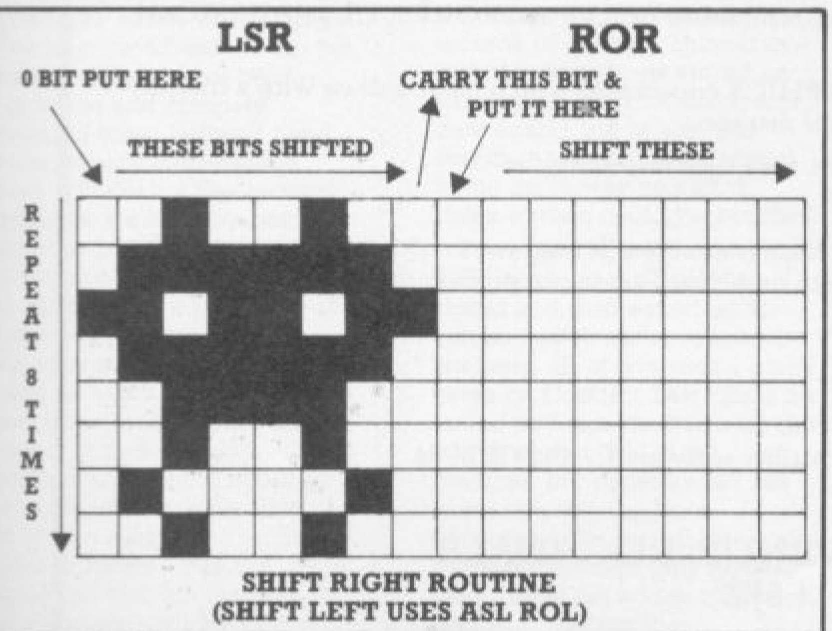
```
LDX #8
LSR 7167,X
ROR 7175,X
DEX
BNE 247
RTS
```

Left shift & to A(7224)

```
LDX #8
ASL 7167,X
ROL 7175,X
DEX
BNE 247
RTS
```

Down (7248)

```
LDX #15
LDA 7167,X
STA 7168,X
DEX
BNE 247
LDA #0
STA 7168
RTS
```



SHIFT RIGHT ROUTINE
(SHIFT LEFT USES ASL ROL)

Right shift A to &(7212)

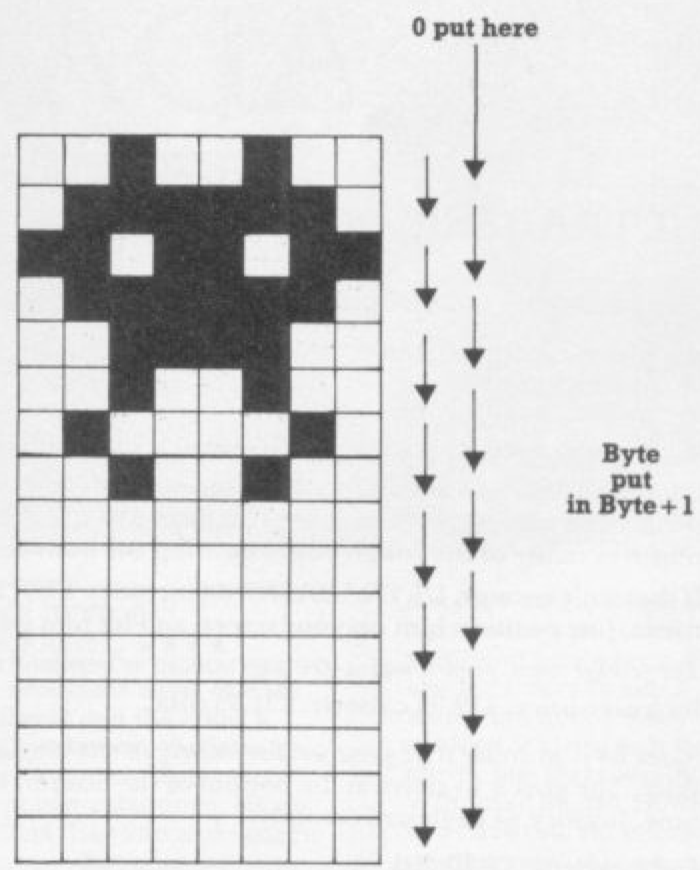
```
LDX #8
LSR 7175,X
ROR 7167,X
DEX
BNE 247
RTS
```

Left shift A to &(7236)

```
LDX #8
ASL 7175,X
ROL 7167,X
DEX
BNE 247
RTS
```

Up (7265)

```
LDX #0
LDA 7169,X
STA 7168,X
INX
CPX #15
BNE 245
LDA #0
STA 7183
RTS
```



DOWN ROUTINE
(UP IN REVERSE)

Now, if you want to really see what even an unexpanded Vic can do, type in the program. It may be fairly long (2.5K) but I can assure you, it is worth it! Games writers now have no excuses

Punters Progress:

Home (and away) — more reviews by J D Collins

Another batch of reviews from JD, this lot being a collection of things that are not games.

Before I turn my attentions to the rest of the programs I must mention a letter from Mark Simpkins (aged 12, he says) pointing out the error of my ways regarding the *Moon Base Alpha* adventure game from Rabbit Software.

Apparently (blush! blush!) it is only *one* game, not two as I thought – which explains a lot, and a lot more than the instructions did. The object of the game is to launch your missile to destroy the meteor which is on collision course with your base. To do this you must find your way through five rooms of the base to collect and use the equipment therein – and do it before the meteor strikes, of course, so you can't spend all day at it! Mark sent me a step-by-step guide to solving the adventure but I am manfully resisting the temptation to use it as that would make it too easy. Any Astronomers out there will be able to report on my progress.

Thank you Mark, keep it up. I was pleased that Mark thought my reviews of the other games were more or less all right, nice to know some one agrees with me about something.

On with the show

Anyway, to this batch of reviews. As you will see most of them are Home Utility things of varying usefulness. I'm in the privileged position of being able to use and compare several of these before I need to fork out any hard-earned zlotys, so I think a few general comments are in order before I get to the individual reviews.

I was specially keen to try the database-type programs as I have been itching to catalogue my Science Fiction books for years now; I have so many I can't remember what I've got in detail. It also occurred to me that I could do the same with my collection of old 78 rpm records.

Unfortunately I was not impressed with the features on the programs reviewed here. That is not to say they don't perform well, it's only that they don't do what I want. Mind you, I have to say that a cassette-based system just can't do the sorting I want: nor can it work at the speed I want the sorting done.

In some ways this would be overcome if all the routines were on ROM so that separate programs for sort and search did not have to be loaded separately – all that messy program changing irritated me no end. I am surprised that someone like Stack with an excellent range of ROM-based utilities has not come up with a database on a ROM. Perhaps there are technical problems involved that make the idea unworkable but I don't see why. Maybe they are waiting for us all to get disk drives!

Come to think of it, I can't see why Stack haven't put a cheap word processor on ROM. With even the memory-less version of Wordcraft 20 costing £80 (twice as much as the ROM-based word processor for the BBC Micro!) there must be scope for competition. How about it, you guys?

The most enlightening (and depressing) thing that came out of this review (and recent articles on database) is that my ideas about a database on a home computer like the Vic are distinctive over-optimistic. I had hoped that a database package would allow me to create records of a length comparable with the documents stored on disk by a word processor, and then enable me to sort and search these quantities of text in the same way as a card index system could be handled.

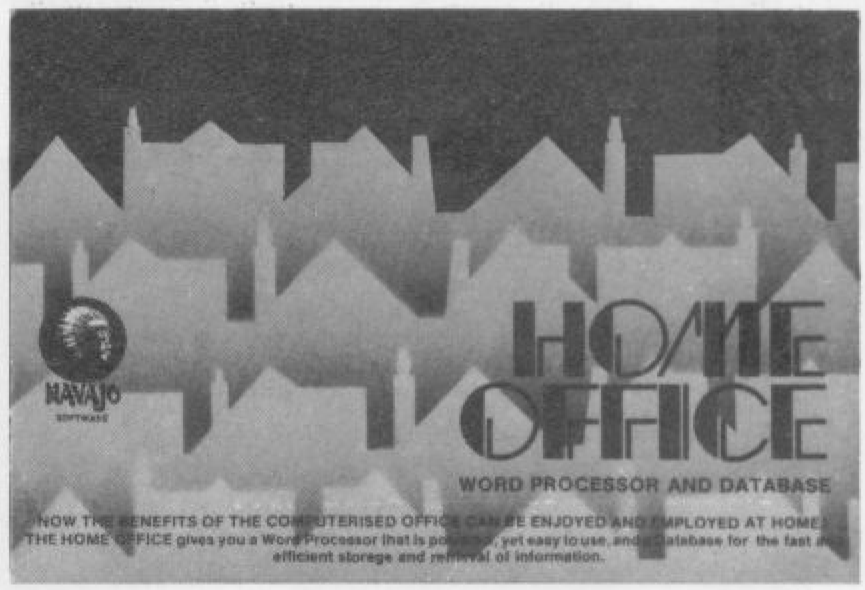
For example, a set of say ten 5,000-word essays could be stored and then searched for certain key items or references. Similarly, all of one term's study notes on Contract Law could be stored and searched in a similar way. It seems this is not possible, but again I can't see why not. Even the encouragingly named *Research Assistant* handled only names, dates, titles not whole chunks of text. Am I asking the impossible? Or is it just that no-one has written the program I want yet?

It seems that it will be some time yet before a really comprehensive range of utilities appears at a price we can afford for the capabilities of the Vic 20.

HOME UTILITIES

HOME INVENTORY
STAR RATING ★★★

Not a game but reviewed here as it is usable on the unexpanded Vic. Useful and easy to use home utility program for recording and valuing your household contents — allows you to update the value of your possessions, too. I grasped the instructions after only a minute's reading of the manual — another plus. Very clear screen display, and can use whatever memory your Vic has. You should think carefully how you will set up your catalogue and it will take some time to get the initial data into the files: but the only real problem is the slow speed and inevitable limitations of the cassette (disk files are so much quicker and easier. . .and more expensive). Cassette; no expansion needed. **Price: £14.95. Supplier: Audiogenic.**



HOME OFFICE
STAR RATING ★★★★

Two programs here, one for databases and the other a word processor. *Vicdata:* for cassette or disk. Enter your data in two catagories, Name and Data. Has Sort and Search features — but only sorts or searches by name, not on data. It's basically a more sophisticated form of VICAT which I reviewed for the February issue of *Vic Computing.* Very easy to use. Clear screen presentation.

Vicpro: modest little word processor, no presumptions but easy on the finger and light on the eye. I was surprised how easy it was to use but I have to admit it would not do for really professional use as it lacks the clever bits that something like Wordcraft 20 has. However if you have the Vic printer for printing our listings and such, this package will make good use of it for letters to pen friends or thank you notes to Aunties for the latest games program you got at Christmas. Don't get me wrong, this is not a put-down: you will find lots of uses for it — but at a domestic level. If you need

NOVASOFT

EXPLOSIVE SOFTWARE
For Vic-20 And Commodore 64





VIC-TORIAN FOLLIES ADVENTURE 1 – SEAQUEST
FOR VIC 20 + 16K EXPANSION
An exciting full colour, multi-screen graphic adventure, set in a romantic era when the foolhardy gentry would risk all, probing the depths for treasure against ridiculous odds.
The ultimate deep sea frolic complete with "Save to Tape" option
Cassette version only £6.95. Disk version only £7.95

DUNGEON DROID
A RACE AGAINST TIME ON A HOSTILE PLANET
FOR VIC 20 UNEXPANDED
Your spaceship is low on Novalite fuel crystals, forcing you to land on a world whose inhabitants worship them! Brave the many dangers of the multilevel tomb and recover enough crystals to take off before the alien lasertanks locate and destroy your ship.
Magnificent machine-code mayhem!
Cassette version only £6.95. Disk version only £7.95.

CYCLOTRON – Fight at the speed of light – A devastating death duel! For the unexpanded VIC 20 and Commodore 64, together on one tape or disk.
CASSETTE VERSIONS ONLY £5.95. DISK VERSIONS ONLY £6.95

JIGSAW – Assembler for VIC 20 + 8K expansion and Commodore 64, together on one tape or disk (details sent on request).
CASSETTE VERSIONS ONLY £9.95, DISK VERSIONS ONLY £10.95

JUGGERNAUT – The memory mover and **MIRAGE** – The memory reconfigurator. Both for VIC 20 (any memory size - details sent on request) Normally £5.95 each.
THIS MONTH ONLY – BOTH FOR ONLY £5.95

NOVASOFT – 46 PHEASANT DRIVE, DOWNLEY, HIGH WYCOMBE, BUCKS HP13 5JL

Please rush me the following:- (All prices include P & P)	TAPE	DISK	**Vic-torian Follies – Seaquest** For VIC 20 + 16K expansion	TAPE	DISK
				£6.95	£7.95
Jigsaw Assembler For VIC 20 + 8K and Commodore 64 on same tape or disk	£9.95	£10.95	**Dungeon Droid** machine code thrills for VIC 20 unexpanded	£6.95	£6.95
Special VIC 20 offer **Juggernaut and Mirage** For the price of one	£5.95	N/A	**Cyclotron (Lazercycles)** For VIC 20 unexpanded and Commodore 64 on same tape or disk	£5.95	£6.95

Name

Address

high-quality text you will pay out a thousand pounds or so for the whole system so we aren't talking about the same sort of thing.

My only reservation on both these programs had nothing to do with their quality; it rests on the limitation of cassette as opposed to disk for this kind of application, an insoluble problem until disk drives plummet in price. Cassette; 8K expansion needed.
Price: £12.95. Supplier: Audiogenic.

RESEARCH ASSISTANT
STAR RATING ★★★

My version was for cassette use only — a disk version is due out soon. This is a set of three programs which allow you to build up, cross-reference and print out research data. Just the sort of thing I think I may need. It allows you to create files of data organised by source code, author, title and bibliography; in those records you may enter notes on the subject as well. Notes are given on how to modify the program to suit your printer. You can edit data, and you can cross-reference by date and by key words. The manual could do with a bit of tidying up but with some messing about I got the hang of it without too much trouble.

Sad to say I found the program itself a bit fiddly to use, though I must stress that this was no fault of the program — it was because of the limits imposed by using a cassette machine to store the data and to load the three programs. Can this type of thing work efficiently without disk? Yes if you can't afford a disk, but it will be an efort. Cassette; 8K expansion needed.
Price: £16.96. Supplier: Honeyfold.

TOTL TEXT
STAR RATING ★★★

A word processor midway between *Home Office* and *Wordcraft 20.* In terms of features it was quite comprehensive for domestic or Club use, say. But I found it harder to use than the other two word processors I have reviewed. Control of the text seemed quite difficult and the manual left a lot to be desired; maybe that was the real problem — with a step-by-step guide like the *Wordcraft* manual it would all have been much easier. Again this was only for a cassette machine (disk version due out soon) and therefore suffered from the inevitable difficulties caused by the slow speed and fiddly usage. Still, quite good for all that. Cassette; 8K expansion.
Price: £16.95. Supplier: Honeyfold.

WORDCRAFT-20
STAR RATING ★★★★★

Our editor lent me his copy to play with and I was much impressed by this word processor — for a fully comprehensive review of all the features on Wordcraft I refer you to the review in *Vic Computing* February 83; all I had time for was a general impression.

The documentation is excellent; it takes you step by step through all the things you need to do to get it running and gives you a "do this, do that, see result" guide to it's capabilities. The friendly tone of the manual was just what a novice like me needs. The control of the colours on screen is fabulous — no more blinding TV screen, no more colour contrast problems! Comes with a cassette for sample documents to load and again you are taken step by step through the ways you can play with (or 'Edit', as it's sometimes called) the text.

I did have a few problems, but they were entirely due to my lack of time to really explore this product; and of course without a printer I could not get a printout of my efforts. I really liked this one. The big question, is do you need all it's features and can you afford the price? Well, if you want to do more than domestic letter-writing you will need something as good as this, and after accounting for the "free" 8K memory you get I think it is reasonably priced. (*There's now a version without the extra 8K of RAM available for around £80 — Ed.*) Cartridge.
Price: £125 (ouch!). Supplier: Audiogenic.



Programming tools

FLOW
STAR RATING ★

On the face of it this seemed a brilliant program: at last you can design your own flowcharts on your Vic. Really useful, eh? Well, it *was* very easy to use (see later!) and the screen display *was* extremely clear. But alas! I could not get any of my 'side chains' or whatever they are called to link up in any way at all. Grrrrrr! I flogged at it for three solid days and then gave up in a temper. Maybe it was me being stupid, maybe it was lack of detail in the manual, maybe it all comes out perfectly when printed (but as I don't have a printer I will never know). Such a disappointment. To be fair, if you can get over this problem I would say it rated four Stars, but I think I'll wait until someone explains it to me. Cassette; 16K expansion needed.
Price: £8.00. Supplier: Computer Room.

PIXEL POWER
STAR RATING ★

An answer to another of my dreams — easy-to-create user-defined graphics. As far as I could tell the program was excellent; easy to use, clear screen presentation, everything I wanted . . . except that I could not get it to modify the graphics on any of my games as it claimed to do. The manual did make a comment about "noting codes for patching in later", but went on to say that to change graphics in a program all you had to do was step A, step B etc. So I did, and it did not work. Of course I tried it several times to no avail, so I can only assume it was not adequately documented. If it can be made to work I would give it a Star Rating of four stars but as with *Flow* I will have to wait for further explanation. Cassette; 8K expansion needed.
Price: £9.50. Supplier: Quicksilva.

MINIKIT
STAR RATING ★★★★★

Really useful utility program: 16 different single-entry functions on function keys plus AUTO (auto numbering of program lines), FIND, DELETE and TRACE (follow your program line by line as it runs). Easy to use, clear instructions, clear screen presentation. Not as comprehensive as the STACK Vickit ROM utility or the Commodore Programers Aid cartridge: but as I have neither (yet) I found this to be a marvellous piece of software. I have details of what the other 'programmers' aids' offer, so I must sit down and work out what I need; maybe MINIKIT will be enough for a sporadic dabbler like myself. I messed about for hours with this and at a fraction of the cost of the other aids it seems excellent value for money — and certainly a good cheap intro to this sort of thing. Cassette; no expansion needed.
Price: £7.95. Supplier: Audiogenic.

Todd's Lore

Todd's Lore

Reconfiguration, logical operators, twiddling bits,



Reconfiguration revisited

The inclusion of the Reconfiguration program in the June **Lore** produced complaints that the Vic didn't work when reconfigured to the +3K option, even though there was 8K or more of expansion memory fitted. Memory errors and errors on loading tapes were reported.

Although this was covered in the article that accompanied the program when it first appeared, the reasons why it doesn't work were only hinted at in the June reprise. Basically, the +3K configuration relies on the existence of a 3K RAM pack which puts memory into the 3K gap between 1024 and 4096 (see the memory maps in the June **Vic Computing** issue). 8K expansion fits in at 8192 onwards, and doesn't fill this 3K gap.

Using the program to set up the +3K configuration assumes that this memory is present – if it's not, all sorts of problems can arise.

Whatever configuration you set up, the program assumes that the correct memory is present, *even if it is not* – but anyone who tries to use it without the correct memory is asking for trouble.

Logical operations

If you program the Vic or the 64, you should be well aware of program lines such as...

IF (A=5) AND (B=6) THEN...

But have you ever wondered what this actually *means*? AND, OR and NOT are 'logical operators' and the use of these, together with an understanding of the way that the Vic and 64 evaluate logical expression can provide us with a useful tool for slightly more advanced programming.

To understand all this needs a reasonable familiarity with binary numbers, as the logical operations of our machines are based on the binary representations of numbers. So a quick bit of revision (a revision of bits?) may be in order.

Raw beginners start here. The binary number system works just like the decimal system, except that we only have the number 0 to 1 to play with instead of 0 to 9. So, counting from zero, we start with 0, then 1 and adding another one gives us 0, carry 1 – just like counting up from 0 in decimal goes from 9 to 0, carry 1.

In some languages such logical variables are specially defined and can be extremely useful. In Basic, we don't need special variables to do this: any Basic variable can be used as a 'flag' in this way.

We could use the logical variable 'OK' as set by the expression above to determine if a number is in the range 0-255 and therefore suitable for POKEing to some location:

```
OK=((X<=255) AND (X>=0))
IF OK THEN POKE 36865,X
IF NOT (OK) THEN PRINT
"ILLEGAL QUANTITY"
```

The NOT function inverts every bit in the logical variable – so that −1 becomes 0 or 0 becomes −1. TRUE becomes FALSE or FALSE becomes TRUE.

The expression can be as complicated as you like, and it can be used to save time where a logical expression may need to be evaluated several times – it just needs to be evaluated once and stored in a logical variable.

The other use is to help make programs easier to read!

Sometimes it is useful to have variables already set to the logical state of TRUE and FALSE, and this can be done at the start of a program by defining these as variables:

TRUE=(1=1): FALSE=(1=0)

Another use for logical variables is where a section of program should be omitted under certain circumstances. For instance, if you've got a section of program that only needs to be executed the very first time the program is run you can use the fact that all variables are initialised to FALSE when a program is RUN:

```
10 IF FLAG THEN 100
20 TRUE=(1=1): FALSE=(1=0)
30 DIM A(100) etc...
40 more initialisation
50 FLAG=TRUE
999 GOTO 10
```

Still with us? If you follow this sequentially, the binary numbers increase from 0 to 1 to 10, 11, 100, 101, 110, 111, 1000 and so on. These are the equivalent of 0 to 8 in the decimal system.

In the decimal system, the 'value' of each place in a number goes from 1 to 10, 100, 1000 and so on. In binary they become 1, 2, 4, 8, 16, 32 and so on, so that 10 in binary is 2+0=2, and 101 is 4+0+1=5.

The fundamental storage location in a computer like the Vic is eight binary digits (or 'bits') in length. That gives 0 as the smallest number which can be stored and 11111111 (which is 128+64+32+16+8+ 4+2+1=255) as the largest.

Now, imagine we have two numbers: let's say 4 (=100 in binary) and 12 (=1100 in binary). We can use the logical operators AND and OR on them quite simply by typing ...

PRINT 12 AND 4, 12 OR 4

You should get '4' and '12'. Why? Because the result of 12 AND 4 (1100 AND 100) has a 1 in the result only where there is a 1 in the first *AND* the second number – the result is 100 (4 in decimal). If we OR the two numbers, we get a 1 in the result where there was a 1 in the first *OR* the second number.

```
12= 00001100    00001100
      AND          OR
 4=  0000010    00000100
    ---------   ---------
    00000100    00001100
      (=4)        (=12)
```

Don't worry about the extra zeros that have appeared – it is usual to show all eight bits when writing bytes out in binary, and the zeroes at the left of the number can be ignored when reading the value. It's like writing the number '12' as '0012'.

Picking at bits

The AND operator provides us with a very useful method of picking out a single bit in a number. The fire button on the joystick controls one of the bits in location 37137 – one of the bits in that location, which is normally set to 1, becomes a 0 when the button is pressed.

In fact, it's the sixth bit from the right – so that, at least in theory, location 37137 is 00100000 if the button is not pressed. Note that we normally number the bits from the right, starting at zero, so the set bit is actually bit 5 of location 37137.

We should therefore be able to test for the button being pressed simply with an IF statement:

IF PEEK (37137)=0 THEN....

...where PEEK(37137) will look at the numeric value of the contents of location 37137.

Unfortunately, many of the other bits of this location are also normally set to 1: and when testing for the fire button we want to ignore these bits.

The easiest way to do this is to 'extract' the bit we're interested in using the AND function. If the button is not pressed, bit six will be set (xx1xxxxx). So you can AND it with 32, which is 00100000 in binary. And the result will be 00100000, decimal value 32. This illustrates the two possible states:

```
PEEK(37137)= xx1xxxxx AND
         32= 00100000
             ------------
              00100000 =32
             (button not pressed)
             xx0xxxxx AND
              00100000
              -----------
              00000000 = 0
             (button pressed)
```

The 'x' indicates that these bits are ignored – because the result of the zero AND another bit is always zero. The result will have a '1' in it only in those positions where there was a '1' in both the first and second numbers.

In other words if the fire button is pressed PEEK(37137) AND 32 would be 0. If not, the result would be 32. This means that the button can properly be tested by:

IF (PEEK(37137) and 32)=0 THEN...

Putting bits back

If AND can be used to extract bits, OR can be used to put them back again! For instance, the three right-hand bits (bits 0,1 and 2) of location 36879 contain the colour of the screen border and are normally set to 011 (*cyan*). to change this to *yellow*, it needs to be 111. This is done as follows:

POKE 36879, PEEK(36879) OR 7

The PEEK (36879) gets the current bit pattern of location 36879, and the OR 7 forces the three bits to 1:

```
PEEK(36879)=xxxxx011 OR
          7=00000111
            -----------
            xxxxx111
```

In the result only those bits are set to 1 where there is a 1 in *either* location 36879 *or* the number 7 – the rest remain unaltered. This is

important as the other bits in this location convey other information regarding the screen and they need to be preserved.

The POKE 36789 puts the result of the OR operation into location 36789 and the colour of the border will change.

This technique can be used to set any combination of bits in a location. But what happens if we want to change the colour to *green*? This has the code 101. We may know how to set bits in a location; but to set the colour green we need to be able to set bits to zero as well.

We can use the fact that with the AND operation, only the bits where there is a '1' in both numbers become '1' in the result – the rest are zero.

So our first task is to clear all the three border colour bits in 36879 as follows:

```
PEEK(36879)=xxxxx111 AND
        248=11111000
            -----------
            xxxxx000
```

Where the number 248 has ones, the contents of 36879 are transferred straight into the result: and where there are zeroes the result has zeroes. Now we are in a position to set the bits we require:

```
Result of above=xxxxx000 OR
              5=00000101
                -----------
                xxxxx101
```

...and the whole command looks like:

POKE36879, (PEEK(36879) AND 248) OR 5

When writing such expressions, it is often a good idea to use brackets fairly liberally to prevent any ambiguity in the line. This is why there are brackets around the AND part of the expression – it's to ensure that this part is executed first. In this example they could be omitted, but it does prevent confusion.

IF...THEN...

Now, let's go back to using logical expressions in IF...statements.

The Vic considers any expression to be TRUE if any of its bits are set to 1. Therefore a statement in the form...

IF 5 THEN PRINT "TRUE"

...will always print TRUE on the screen, because 5 has two bits set to 1. FALSE is indicated by all zeroes: and so the value 0 would result in the PRINT statement never being executed.

Normally, when evaluating a logical expression, the Vic and 64 will set all the bits to '1' if it's TRUE and all to zero if FALSE. So, typing...

PRINT &=6

...will result in 0 being printed, while...

PRINT 5=5

...will result in (wait for it!) –1 being printed.

The computer always does its logical operations as if they were integer variables – that is, 16-bit binary numbers: and it so happens (for reasons I'll explain in a moment) that a number in the form 1111111111111111 is actually –1.

This also means that we can just as well set a variable to the result of a comparison, so that...

OK=((X<=255) AND (X>=0))

...is a valid expression, and the variable OK will be set to –1 (TRUE) if X is less than or equal to 255, AND it's not negative; and it will be set to 0 (FALSE) if it's not.

First time through, FLAG is false; so the branch to 100 in line 10 is not taken and the initialisation is performed. FLAG is then set to TRUE. Subsequent execution of line 10 will always result in a jump to line 100, thus avoiding reinitialising – unless you deliberately reset the FLAG to FALSE.

Sometimes it is necessary to execute a section of program on alternate passes. Simply use a FLAG variable, and in the loop have this line:

FLAG=NOT(FLAG)

Then you can test for FLAG at the start of the loop – it will alternate between TRUE and FALSE.

65535=–1

Why should a variable set to all ones be –1? Well, using 16 bits (which is, of course, two bytes) we can have a range of numbers from 0 to 65535. But they're all positive, with 65535 being 16 ones.

To cope with negative numbers we have to consider what happens as we count down from 32767. In decimal this would be 32767, 32766...4, 3, 2, 1, 0, –1, –2 and so on. In 16-bit binary it would be:

```
0111 1111 1111 1111 (32767)
0111 1111 1111 1110 (32766)
0000 0000 0000 0010 (2)
0000 0000 0000 0001 (1)
0000 0000 0000 0000 (0)
0000 0000 0000 0000 (-1)
1111 1111 1111 1110 (-2)
1000 0000 0000 0001 (-32767)
1000 0000 0000 0000 (-32768)
```

If you don't believe that the all ones situation is –1, just try adding 1 to it and see if you get zero (ignoring the 17th bit, as there's nowhere to put it.)

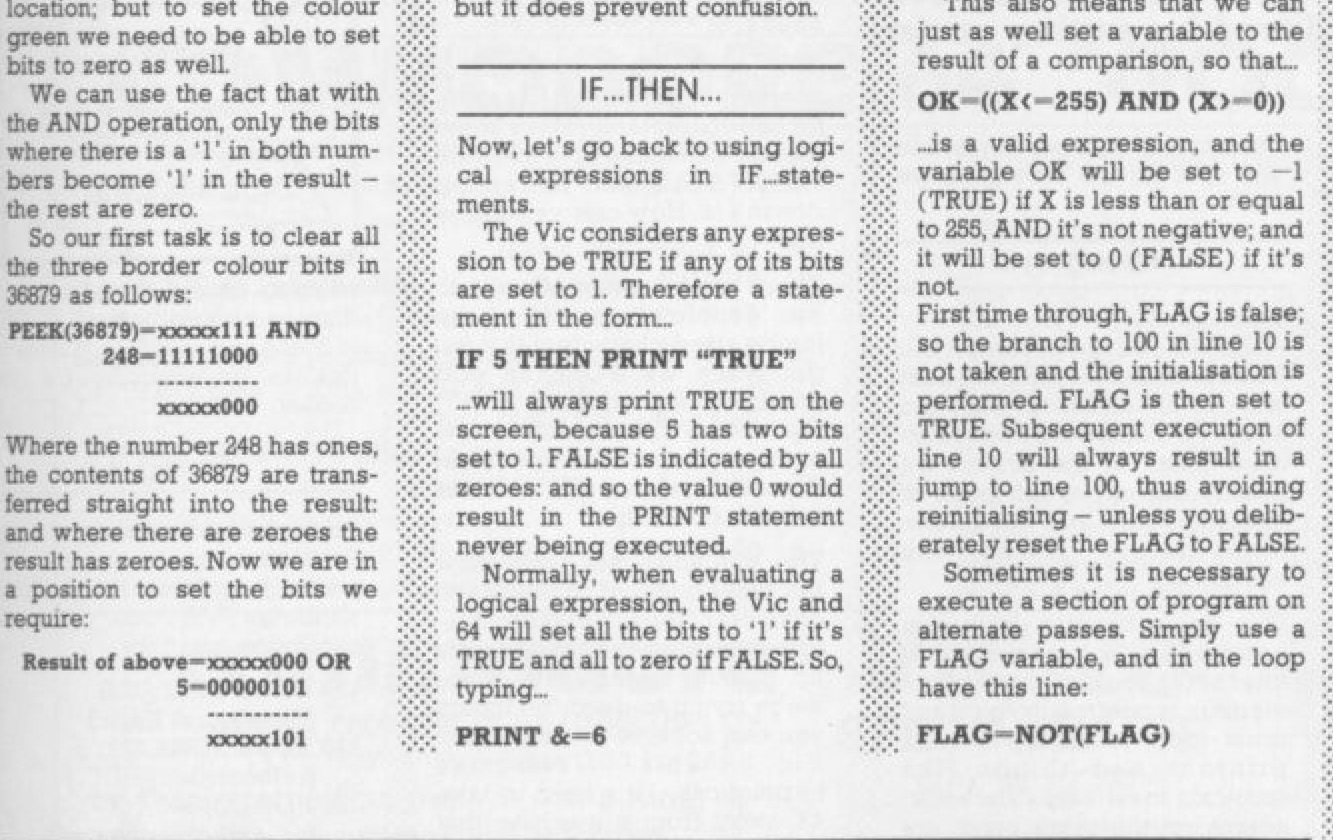
What we now have is a situation where we can cope with negative numbers and in this case, the 16th bit is no longer part of the number – it becomes the sign of the number (1=negative). This means that the upper limit of positive numbers is 32767 andd the lower limit is –32768, which is exactly what the limit of integers on the Vic actually is!

Numbers can be converted to their negative value by inverting all the bits and by adding 1. Try this with the values in the above table.

You should now be able to see why the value of TRUE is –1, and FALSE is 0.

Of course, the interpretation of negative numbers in this way is purely arbitrary. And integers with their 16th bit set to 1 could be read as greater than 32768; it's simply a case of how the Vic has chosen to interpret the number.

For instance, try **POKE 56,255** and then **PRINT FRE(0).** On a completely unexpanded Vic, you should get –4355. What you've done is told the Vic that it's got more than 32767 bytes available (61180 to be precise). When it then tries to return this value as a number, the fact that it is greater than 32767 (with bit 16 set to "1") makes it assume that the number is negative: and the 61180 is interpreted as –4355. ■



Visiting Vic-20 Video Part 4

by Jim Butterfield

In which the traveller finds that the highest resolutions can be achieved by setting one's sights a little lower...

We've spent some time viewing the world (or at least memory) from chip's-eye view, and have noted that the video chip sees memory in its own way:

We've muddled with the character set, both built-in and home-brewed. But we haven't seemed to deal with achieving that mystic goal — high-resolution screen control.

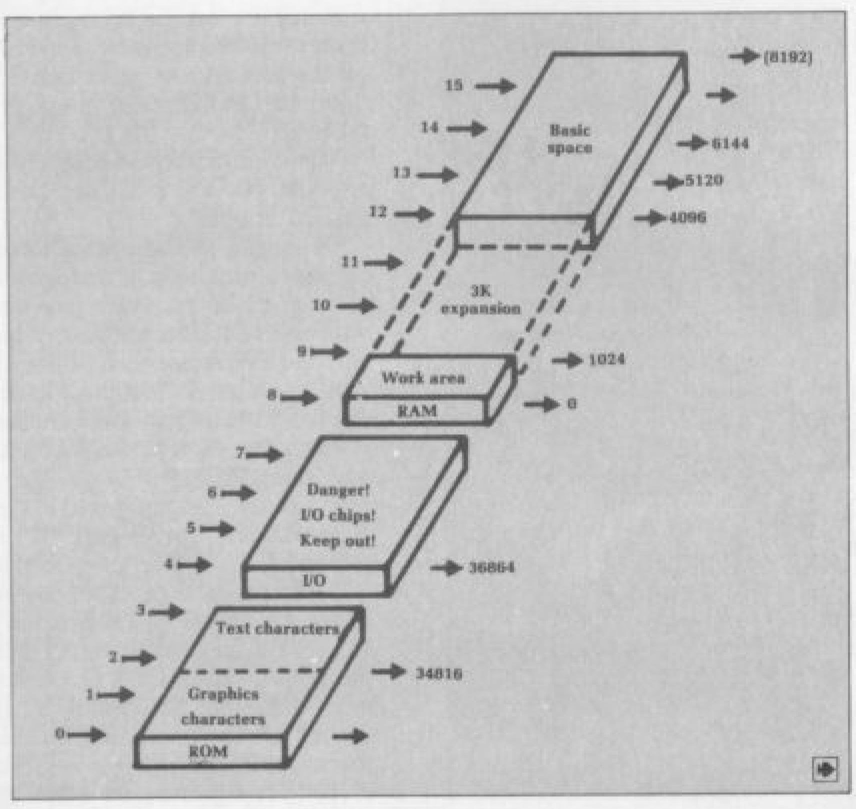
We've dealt with custom characters. And as Glinda the Good Witch almost said to Dorothy, "If you had known their powers... you could have done it the very first day". In other words, we've been looking at high resolution all along without recognizing it.

Here's the trick: if every position on the screen contained a different character; and if we can define any character at will; we can define any spot on the screen as we wish.

Filling in

Mechanically, we do it this way — the first cell on the screen will contain character zero, the next will contain character one, and so on. To change the upper-left-most pixel on the screen we modify the upper left pixel of character zero and the screen immediately shows the alteration.

This is a change from our normal use of screen and character set. Our screen memory is now totally fixed and must not change. Normal printout and things like scrolling must stop. The characters, on the other hand, are now completely variable, with pixels turning on and off according to what the picture needs.



Wait — there's a problem. It seems that the screen has room for 506 characters; yet we know that we can make only 256 individual characters. Something doesn't fit. How can we resolve this?

There are two ways. One is to use 'double characters' — the jumbo sized characters that we get when we POKE an odd number into address 36867. Each of our 256 characters now occupies twice the space on the screen, so that we can cover the screen easily.

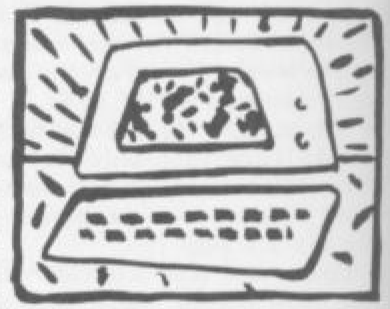
The character set table now becomes huge, of course: each character takes 16 bytes to describe, making the whole table up to 4096 bytes long. Since we're trying to describe things you can achieve in a minimum Vic, this becomes impractical... it's hard to take 4K away from a machine that has only 3.5K available to start with.

On a machine with memory expansion, however, this is quite practical; read on, for we'll use tricks on the small machine that will come in handy even on the big ones.

The other method is this: cut the size of the screen so that it contains only 256 characters or less. We can store the number of columns and rows we want into 36866 and 36867.

POKE 36866,16 will set sixteen columns; and POKE 36867,32 will set sixteen rows (we must multiply the number by two here). How many characters can we store? 256 characters, that's how many; and that number may sound familiar by now.

By the way, Basic won't know how to cope with the peculiar row and column counts if you do this as a direct command. So be prepared for an odd-looking screen. Neatness fanatics will want to centre the remaining display by appropriate POKEs to 36864 and 36865, but I'll leave this as an exercise for the reader.



Doing it

Enough of this abstract theory. Let's dive into a program to prove that even the humble minimum Vic can do high-resolution graphics.

```
100 POKE 56,22:CLR
    (drop top of Basic)
110 POKE 36869,222
    (relocate screen...)
120 POKE 36866,144
    (...and character set)
```

Note that the above line sets the screen to a half-block (128) and sets up 16 columns instead of the normal 22 (128 plus 16 gives 144). We may as well go ahead and change the rows:

```
130 POKE 36867,32
    (16 rows times 2)
200 FOR J=6144 to 8191
210 POKE J,0: NEXT J
```

We've cleared the entire character set to zero (all pixels off). Now let's set up the screen with character zero in the first slot and so on:

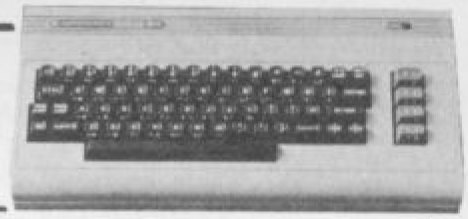
```
300 FOR J=0 TO 255
310 POKE J+5632,J
320 NEXT J
```

Let's set all characters to colour black:

```
330 FOR J=37888 TO 38911
340 POKE J,0: NEXT J
```

Our screen is now ready. Serious graphics takes quite a bit of maths (dividing by 16 to find the row and column; dividing by 8 for the pixel posi-

4 SPECIAL OFFER PACKAGES



1. commodore 64

+ cassette deck + dust covers p&p code

if purchased seperately these items would cost £278.85

OUR PRICE **£255.00** d

2. commodore 64 + cassette deck

+ Introduction to Basics Part I (Manual and 2 cassettes to take you through the first steps in computing) + DFM Data Base (Information Storage Programme) + 2 dust covers. If purchased separately these items would cost £313.00.

OUR PRICE **£283.00** d

3. commodore 64

+ VIC 1541 disk drive

+ Box of 10 diskettes + dust covers. If purchased separately these items would cost £490.00.

OUR PRICE **£435.00** d

4. Businessman's Starter System

Comprises Commodore 64 + VIC 1541 disk drive

+ box of 10 diskettes + VIC 1525 Printer + Box of paper + 3 dust covers. If purchased separately these would cost £735.00

OUR PRICE **£660.00** d

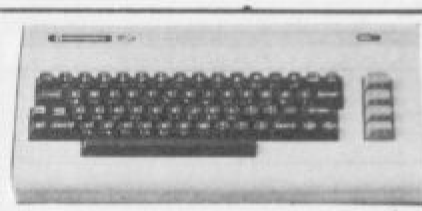
COMMODORE 64 "GOODIES"

NEW 64 Printer VIC 1526 with cable	£345.00	d
NEW Printer/Plotter (4-colour)	£169.99	d
NEW LOW PRICE — Disk Drive VIC 1541 with cable	£229.00	d
Printer VIC 1625 with cable	£212.75	d
Book: 'Programmer's Reference Guide'	£ 9.95	b
Easy Script (Word processing package)	£ 75.00	b
DFM Data Base — Disk or Cassette (state which required)	£ 19.50	b
DFM Mail labels (Print labels to your specification	£ 9.50	b
Catalogue (Dedicated filing system)	£ 16.50	b
Datalog (User orientated filing system)	£ 17.50	b
Journal (A financial diary with permanent	£ 19.50	b
Gortek & Microchips (Designed to teach 10/13 year olds basics)	£ 12.95	b
Sales Ledger	£113.85	b
Purchase Ledger	£113.85	b
Easy Stock	£113.85	b
Attack of the Mutant Camels (Try and beat hostile Aliens)	£ 8.50	a
Motor Mania (A thrilling cross country car race)	£ 8.95	a
Dust cover for 64/VIC 1525	£ 2.95	a
Joystick	£ 6.90	b
Introduction to Basics Part I (see above)	£ 14.95	b

Many other titles in stock. Free list with first delivery. Or send S.A.E. for full list.

commodore VIC 20

SPECIAL OFFER PACKAGE



+ C2N cassette recorder + Introduction to Basics Part I (Manual and 2 cassettes to take you through the first steps in computing) + I cassette with four games + Only from Chromasonic. FREE DUST COVER normally sold for £2.95. If purchased separately these items would cost £249.99

OUR PRICE **£139.99** d p&p code

VIC 1525 Printer with cable	£212.75	d
VIC 1541 Disk Drive with cable	£229.00	d
16K RAM pack	£ 28.95	b
24K RAM pack	£ 47.95	b
Joystick	£ 6.90	a
4-slot motherboard	£ 33.35	c
Dust cover for VIC 20/VIC 1525	£ 2.95	a
Light pen and free game	£ 28.75	c
Super Expander (High resolution cartridge with 3K RAM)	£ 31.00	b
Programmers Aid cartridge (Adds more than 20 commands)	£ 28.00	b
Machine Code Monitor (Gives you 6502 assembly language)	£ 28.00	b
C2N cassette unit with cable	£ 39.10	c
Sargon Chess (6 levels of play)	£ 23.00	b
Choplifter (Rescue your men and then return to base)	£ 24.95	b
VIC Programmer's Reference Guide	£ 9.99	b

Many other titles in stock. Free list with first delivery. Or send S.A.E. for full list, state which system required.

Our Guarantee—1 full year for labour and parts on all Hardware. Prices correct at time of going to press

CHROMASONIC PERSONAL COMPUTERS

48 Junction Road, Archway, London N19 5RD

TEL: 01-263 9493 or 5

a = 60p
b = £1
c = £2.50
d = £5

- FREE ADVICE on all aspects of Personal Computers. You only have to ask.
- DELIVERY: On P.O.'s, Access or Barclaycard, dispatch usually within 24 hrs. Or we'll advise by return. On cheques allow about 5 days.
If not completely satisfied, return the goods to us undamaged within 14 days for a full refund including p&p. VAT included in all prices.

ACCUMULATOR DISCOUNT COUPONS
A Chromasonic special

On every £5 worth of Software you buy from us, over the counter or by mail, we'll give you a 50p ADC. So with a £20 purchase, you get four ADC's etc. The same applies to hardware, but here, we give you a 50p ADC on every purchase of £10, or multiples of £10. And that goes for virtually everything we sell. Phone or write for further details.

tion...). But we'll substitute a little simple coding to draw a traingle:

```
400 FOR J=6792 TO 6816
    STEP 8
410 POKE J,255 (horizontal
    line)
420 NEXT J
500 FOR J=6280 TO 6664
    STEP 128
510 FOR K=J TO J+7
520 POKE K,128 (vertical
    line)
530 NEXT K,J
600 FOR J=6280 TO 6704
    STEP 136
610 X=128 (leftmost pixel)
620 FOR K=J TO J+7
630 POKE K,PEEK(K) OR X
640 X=X/2 (move pixel
    right)
650 NEXT K,J
700 GOTO 700
```

The program is now complete. It will wait in a loop at line 700 until you press RUN/STOP... when you do so, a number of odd things will happen. The computer will try to print the word READY into screen memory... but screen memory is intended for a different usage now, and all that will result is screen clutter.

Bring everything back to sanity by holding down RUN/STOP and tapping RESTORE.

Extra ideas

Effective graphics call for a fair bit of arithmetic. To place or clear a pixel, you need to find the row and column by dividing the X and Y co-ordinates by the appropriate scaling factor.

You need to change this to a screen character number by multiplying the row number by the total number of columns and then adding the column number. Multiply this by eight, and you'll get the position where the character is located within the character set.

Now we must go for the pixels within this character: the bits within a byte are pixels 'across' and the eight consecutive bytes are pixels 'down'. Now you know why people buy a Super Expander: to save them from the maths!

Even when you have plenty of memory available, which allows you to use double characters and get lots of pixels on the screen, it's usual to trim the screen a little.

The normal 22 columns by 23 rows are usually trimmed back to 20 columns by 20 rows (actually 10 rows of double characters).

This does two things: it makes the arithmetic a little easier, and it drops the memory requirements from 4096 bytes for a full deck down to only 3200 bytes. This, in turn, gives us space to pack screen memory into the same 4K block. That's handy because we cannot be sure that the video chip will have access to any more than 4K of RAM. Basic, of course, will long since have been moved to occupy memory from 3192 and up.

If you want to add text to the high resolution display, it's a snip. Just copy the characters you want from the character set ROM and transfer them to the appropriate character slots on the screen. (Of course you would have thought of that yourself if I hadn't just told you.)

Don't forget that you can POKE appropriate values into 36864 and 36865 to centre the graphics neatly. Our example looked a little lopsided: try your hand at making it neater.

High resolution is there and waiting. There's a bit of mathematics involved. You may find this a challenge: after all, isn't that what a computer does best?

Even if the maths befogs your mind and causes you to go out and buy a Super Expander, you'll have learned a few new things. First, the Super Expander doesn't make graphics possible — they were there all the time — it just makes them easier. Secondly, you'll have a better idea of what's going on inside your marvellous computer.

THE COMPLETE BEGINNERS GUIDE *TO ASSEMBLY LANGUAGE PROGRAMMING*

The second in a series of articles to lead you gently into the world of 0s and 1s with the Vic-20.

IF YOU READ THE FIRST ARTICLE in this series, which appeared back in June, you will now be eagerly waiting to learn the first instructions in 6502 assembly language programming and run your first machine-code program.

No, you don't need Vicmon yet: we are going to POKE the program into memory. Type in and RUN this Basic program on your unexpanded Vic exactly as shown, and see what happens (hmm ... that sounds familiar):

```
10 PRINT "CLEAR HOME"
20 POKE 56,28 : POKE 52,28 : CLR
30 DIM A(20)
40 FOR I = 1 TO 20
50 READ A(I) : POKE 7199+I,A(I)
60 NEXT I
70 DATA 169,4,162,26,157,7,151,202,208,250,162,26,138,
   157,7, 31,202,208,249,96
```

Nothing appears to have happened ... yet. Now type **SYS 7200.** You should see all the letters of the alphabet appear in the middle of the screen. I shall now proceed to explain how your 6502 accomplished this amazing feat; you have, in fact, just run a machine-code program.

The 6502 registers

Inside the 6502 there are a number of registers:

- Accumulator
- X index register
- Y index register
- Status register
- Stack pointer
- Program counter
- Instruction register

There is also a 'black box' called the arithmetic and logic unit (ALU) – we don't need to understand how it works, only what it does.

Each 6502 instruction is one, two or three bytes long. The instructions (the program) and the data all reside in memory. When a program is RUN, each instruction is fetched in turn from memory into the 6502; there it is decoded and executed. The program counter always contains the address of the next instruction to be executed; it therefore has to be 16 bits wide in order to specify an address (range 0 to 65535). All the other registers are eight bits wide.

The instruction is fetched into the instruction register for decoding; at the same time the program counter is automatically incremented to point to the address of the next instruction. The contents of the instruction register are decoded by the ALU, which then carries out the instruction.

The program operates on data. Most operations require that the data resides inside the 6502 while the instruction is executed. To this end certain working registers are needed to hold the data, the accumulator being the primary one.

To give an example, take the addition of two eight-bit numbers. One of them is first fetched into the accumulator. This is then added to one residing in memory, and the result deposited in the accumulator. This result will now have to be stored back in the memory ready for the accumulator to accept subsequent data. The X and Y index registers are most often used as counters, as shown in the next sample.

Locating your program

Now, the adding of two eight-bit numbers is pretty dull: and we can do much more interesting things to get you started.

First things first, though. You have to decide where you are going to store your machine-code program in the VIC's memory. While experimenting, my preference is to use the unexpanded Vic and lower the top of memory available to Basic by 512 bytes from location 7680 to 7168. This can be accomplished by entering the line:

POKE 56,28 : POKE 52,28 : CLR

You need not understand how this works, only that it sets aside 512 bytes of memory for your machine-code program and prevents it being overwritten by Basic. I then start the program at location 7200, and waste 32 bytes!

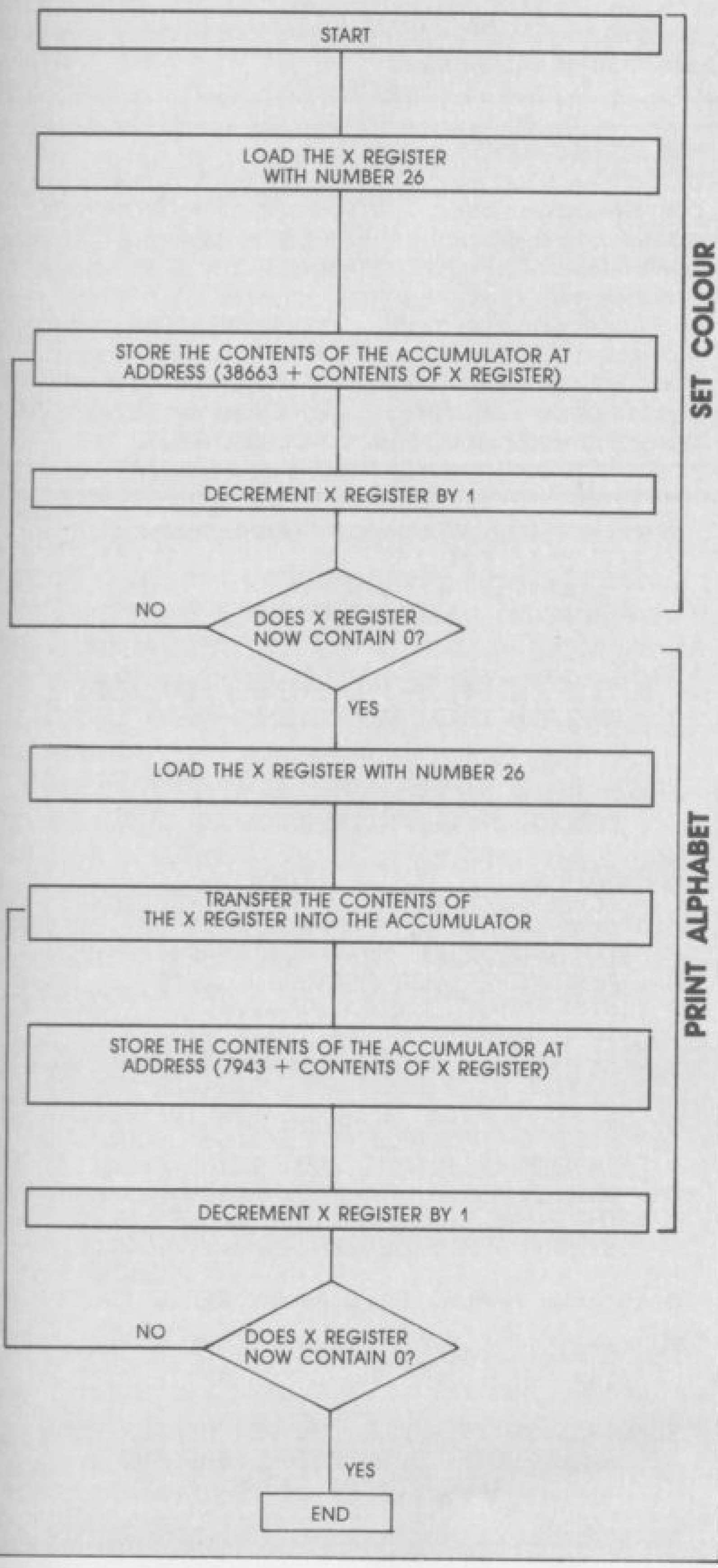
Program objectives

Next, you should decide on the objectives of your program. The objectives of the alphabet program are to place screen codes of 1 to 26 in locations 7944 to 7969 respectively (somewhere in the middle of the screen) and a colour code of 4 (purple) in locations 38664 to 38689, the corresponding colour RAM addresses.

Diagram 1 is a flowchart illustrating the various program steps which will achieve these objectives. If you're impulsive like me, you won't bother with the flowchart; everyone else will stress how important it is, and they're probably right.

DIAGRAM 1

Flowchart for Alphabet Program



Program coding

The final step is to code your program as shown in **Diagram 2**. Each section in the flowchart corresponds to one instruction, and you'll need to be able to look up the instruction to obtain its code. (A full list of 6502 instructions will appear in a subsequent article in this series: but if you can't wait, refer to the *Programmer's Reference Guide* or a 6502 Assembler book like Leventhal's.) For example, 'LOAD THE ACCUMULATOR WITH NUMBER 4' has an instruction code of 169 (decimal) followed by the number 4.

There are several ways to load your program – the method used in the example at the beginning of the article was to load it from within a Basic program (you'll see that the numbers in the DATA statement correspond to those in the 'decimal contents' column of Diagram 2) and then type **SYS 7200**, the program start address.

Assembly language

Coding the program as in Diagram 2 results in a series of numbers which are pretty hard to decipher. The right-hand column contains the three-letter 6502 mnemonics, though, and these are far more meaningful to the program reader.

These assembly language instructions can be entered directly if you possess Vicmon or a similar assembler; otherwise it's still probably better to write your program in 6502 assembly language first, and then convert it into numbers yourself afterwards.

To finish off

So far I have not described the functions of the status register or the stack pointer; the latter I'll leave until next time. The status register consists of eight independent bits, each of which is used as a flag which is either set (1) or clear (0):

DIAGRAM 2

Machine Coding the Alphabet Program

ADDRESS		CONTENTS		VICMON ASSEMBLY INSTRUCTION
DECIMAL	HEX	DECIMAL	HEX	
7200	$1C20	169	$A9	LDA #$04
7201	$1C21	4	$04	
7202	$1C22	162	$A2	LDX #$1A
7203	$1C23	26	$1A	
7204	$1C24	157	$9D	STA $9707,X
7205	$1C25	7	$07	
7206	$1C26	151	$97	
7207	$1C27	202	$CA	DEX
7208	$1C28	208	$D0	BNE $1C24
7209	$1C29	250	$FA	
7210	$1C2A	162	$A2	LDX #$1A
7211	$1C2B	26	$1A	
7212	$1C2C	138	$8A	TXA
7213	$1C2D	157	$9D	STA $1F07,X
7214	$1C2E	7	$07	
7215	$1C2F	31	$1F	
7216	$1C30	202	$CA	DEX
7217	$1C31	208	$D0	BNE $1C2C
7218	$1C32	249	$F9	
7219	$1C33	96	$60	RTS

Bit	Name	Function
0	CARRY	is set when the result of an operation would have more than eight bits
1	ZERO	is set when the result of an operation is zero
2	INTERRUPT	is set by the processor each time there is an interrupt
3	DECIMAL	is set if the 6502 is to perform binary-coded decimal arithmetic
4	BREAK	is set by the processor each time there is a programmed break
5	UNUSED	
6	OVERFLOW	is set when the result of an operation is too large, usually signed binary
7	NEGATIVE	is set when the result of an operation is negative

These flags play a very important role in programming, and we'll need to understand all of them eventually. For now I'll just explain two of them.

First, the **zero** flag. It is set to 1 automatically by the 6502 each time the result of an operation is zero. The BNE (branch if not equal to zero) instruction in diagram 2 tests this flag; and depending on whether it is clear or set, it will or will not branch elsewhere in the program.

Second, the **decimal** flag. It is up to you, the programmer, to decide if you wish to work in binary-coded decimal (BCD) or binary. To communicate your desires to the 6502, the assembly language instruction SED (set decimal) will cause it to operate in BCD mode; CLD (clear decimal) will cause it to operate in binary mode.

And what is BCD anyway, you might ask? It is quite simply the encoding of a single decimal number (from 0 to 9) in four binary digits. For example, 00110010 in binary means '50'; in BCD you read the 0011 (three) as the first digit and the 0010 (two) as the second, giving thirty two. Obviously, codes in excess of nine (like '11', coded in binary as 1011) are illegal in BCD and must not be used.

Next time we'll take a closer look at some assembly language instructions and something called addressing technique. Meanwhile, maybe you can display some other characters on the screen?

Classified

Graphics Packages Review

by Mike Todd

Last time I looked at Programmers aids – collections of utility commands to make the task of programming a bit easier and designed specifically for use during the development stage to help in writing and debugging the programs.

Extensions to the Basic language are also available to help programmers. These are intended to eliminate the large number of PEEKs and POKEs, convoluted programming and inside knowledge needed to perform such apparently simple tasks as plotting points on the screen, drawing lines or even just changing the colour of the screen border.

The disadvantage with this sort of package is that not only must it be operating *while the program is being developed* but also *when the program is being run.* Portability of the software therefore suffers as it can only be run on suitably equipped machines.

Both Commodore and Stack have produced such packages, both of which require a minimum of 3K expansion, but will work with 8K or more. The Commodore Super Expander is a self-contained cartridge, complete with the extra 3K RAM required: it costs £34.95; the Stack VICIT III is available as an EPROM chip at £21.85 (which will require a board with a suitable EPROM socket) or as a cartridge for £33.35, but you will need the extra RAM. Stack also has available the Supercharger Plus, which contains the VICKIT I Programmer's Aid, VICKIT III and 3K of RAM for about £56.

Because the Vic requires its memory to be reorganised to enable high-resolution graphics to be used, a command is needed to set this up.

VICKIT III has the command **GRAPHICS** which does the reconfiguration. As a result, it also clears all variables as soon as it is used: and therefore it's used only once, at the start of a program using graphics.

The graphics screen is selected with **CLEAR x** which sets the character/plotting colour according to x and clears the graphics screen. **SCREEN x** does the same, except that it *doesn't* clear the screen – which has the advantage that you can return to a previously set up graphics screen without it being cleared.

TEXT UPPER (or **TEXT LOWER**) will return to a cleared text screen.

The Super Expander uses a single **GRAPHIC x** command, where x determines the graphics mode. If x–2, the high-resolution mode is set: x–0 returns to the normal text screen.

The command can be used anywhere in a program; but as with VICKIT III first time the grap' ics mode is selected it clears all variables. So it should appear at the very start of aprogram, even if this only selects the high-resolution screen and the program then restores the text screen.

Mode 1 is the so-called 'multicolour mode' (medium resolution). Mode 3 is effectively a mix of modes 1 and 2, with the actual plotting mode being determined by the character colour in the same way as on the VICKIT III.

Because of the limitations of the Vic's memory, neither package can use the whole of the screen; and the Super Expander reduces the 'window' slightly both horizontally and vertically. It gives a total capability of 160 x 160 plotting points.

The VICKIT III sets the screen to 176 x 176 points, but it will not allow the points to be plotted on the bottom two character lines – which gives the same vertical resolution of 160 points with a slightly wider horizontal capability of 176 points.

Each point on the screen is accessed using x, y coordinates (x is the horizontal position and y the vertical) with the VICKIT III running from 0,0 at the bottom left to 175,159 at top right.

The Super Expander is rather different in that 0,0 is actually at the *top* left, and the bottom right is 1023,1023. This means that the screen consists of an imaginary 1024 x 1024 points, and all coordinates need to be calculated assuming that this is the case. When it comes to plot the point, the Super Expander 'rounds off' the co-ordinate to the nearest values in the range 0 to 159.

This may seem an odd thing to do, but it allows programs written using the Super Expander equivalent as standard, it would have meant Vic graphics programs could have been used on the 64. They could have taken advantage of the better resolution without any software changes. But it hasn't and so they can't).

Colours

With high-resolution characters on the Vic, there is a limited number of colours that can appear on the screen.

The graphics screens are actually made up from areas or regions, each 8 dots wide by 16 dots deep, and in this region the dot colour must remain fixed. Plotting a point of a different colour will change all points in that region to the same colour – the character colour.

In high-resolution mode, the only other colour that affects the graphics is the background colour. This means that in these mode, only two colours are definable in each 8 x 16 region. Although the character colour can be different in different regions, the background colour remains the same over the entire screen.

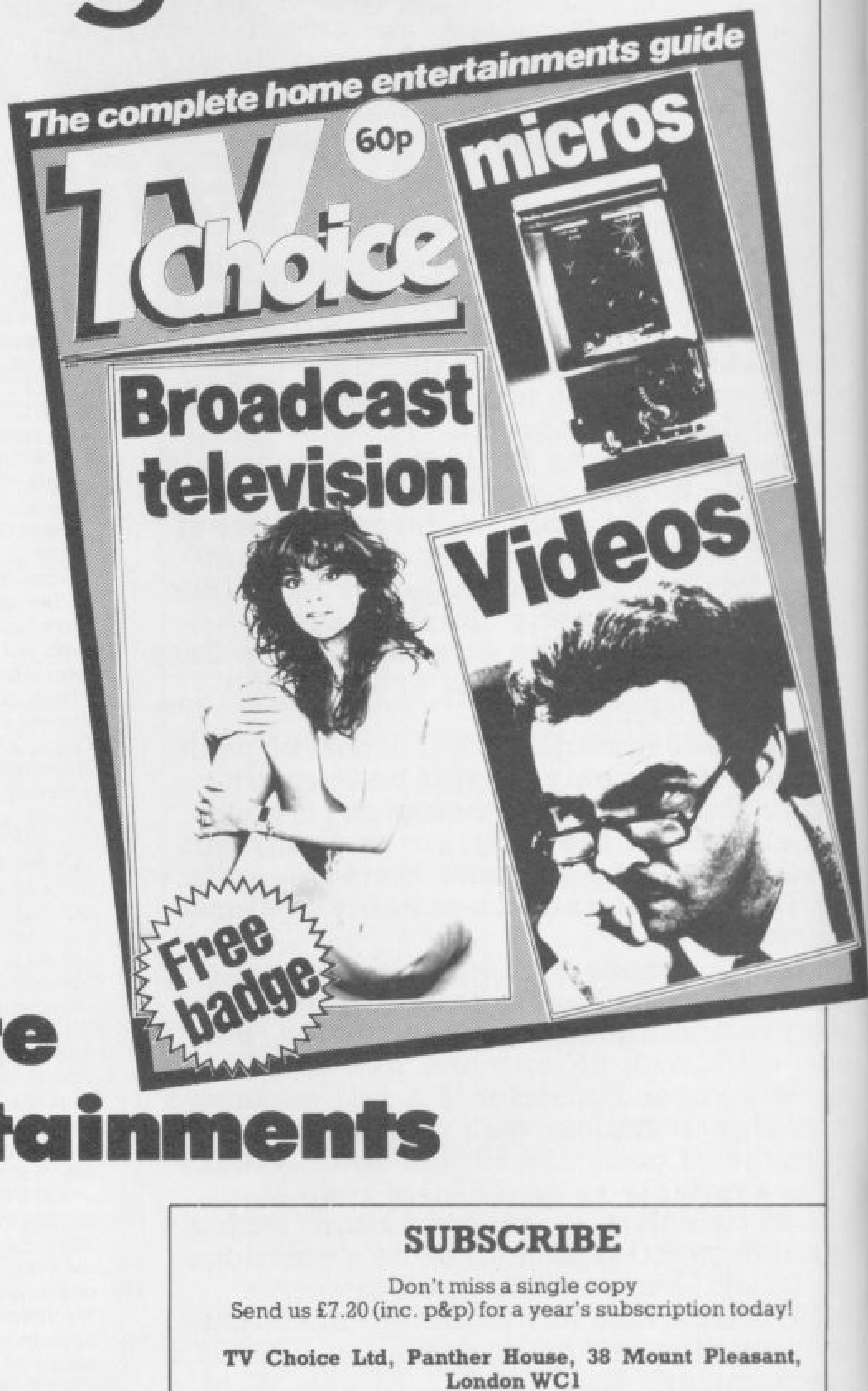
The medium-resolution (multi-colour) mode allows four definable colours in each region, but this results in a halving of the horizontal definition.

In this mode, the four colours are defined as the *background* colour, the *border* colour, the *character* colour and an *auxilliary* colour. The border colour dcan be set at any time, but in affects only the graphics on the screen in multi-colour mode; so does the auxilliary colour-

The Super Expander has two commands for setting the colour

The revolution has begun!

The new TV age is here. The days when watching TV was a case of choosing between three broadcast channels are over. The revolution has begun. The TV revolution. Video, video games, home computers, cable, satellite: It's a maze of confusing, exciting options. You need a magazine to help you through the maze, a magazine that tells you *everything* you can do with your TV. TV Choice is out every month. Order your new-look TV Choice now, only 60p.



The complete home entertainments magazine!

TV Choice

SUBSCRIBE

Don't miss a single copy
Send us £7.20 (inc. p&p) for a year's subscription today!

TV Choice Ltd, Panther House, 38 Mount Pleasant, London WC1

Here's my £7.20. Enter my subscription today.

Name. . . . .

Address . . . .

. . . .

. . . .

registers. The first is **COLOUR s,b,c,a** – where **b** is the border, **c** the character and **a** the auxiliary colour. Obviously, in high-resolution mode, only **s** and **c** have any affect on the plotted points. **REGION c** is a command which changes the character colour during graphics work.

The VICKIT III has the separate commands **BORDER** and **GROUND** for changing these colours, and the **CLEAR** or **SCREEN** commands are used to set the character colour. There appears to be no command for setting the auxiliary colour.

Dots on the screen

The main function of the two packages is to allow you to put dots on the screen without resorting to complicated PEEKs and POKEs. VICIT III has commands to **SET, RESET** and **INVERT** points; the Super Expander just has a **POINT** command.

In fact there is a difference in philosophy between the two packages. VICKIT III always allows you to decide whether points should be set, reset (back to the background colour so you can't see it) or inverted, and this applies to all its graphics commands. As there is no built-in facility for using the multi-colour mode, all points are set to the colour defined by the CLEAR or SCREEN commands.

The Super Expander, on the other hand, requires you to specify which colour register you wish to use when plotting points. Therefore **POINT 2, 50,50** will set a point at co-ordinate 50,50 using colour register 2, the character colour. Register 0 is the background colour; and so to reset or clear a point, you plot using register 0.

Only registers 0 and 2 are available in high-resolution mode. But provision *is* made for using the multi-colour mode, with registers 1 (border colour) and 3 (auxiliary) being available.

In these cases, it is the *colour register* and not an actual colour that is used to plot points; and so, using the COLOUR command, it is possible to change these colours at any time – even after the points have been plotted.

This 'palette' technique can be very useful in creating exciting colour graphics, and only the Super Expander is really geared up to do this.

Beyond the single point

Of course, single points can be plotted in sequence to generate lines, boxes and circles: and this can be done in Basic – but it is very slow.

Both packages have a repertoire of line-drawing commands, all using the same basic philosophy as the single-point commands. This can both draw straight lines from one co-ordinate to another, with the VICKIT III being additionally capable of drawing a rectangle having been given the co-ordinates of opposite corners (although the SUPER-EXPANDER can take a series of co-ordinates in the same command and join these points).

Both allow circles (or parts of circles) to be drawn, although the slightly squashed screen format makes these rather oval. Both packages can however alter the 'shape' of the circles tomake them more round, but only the Super Expander does this with any real flair; VICKIT III has very severe limitations on the ratio of height to width.

Plotting circles (and lines for that matter) takes quite a while – about a second for a screen-sized circle – and there is not much to choose between the packages in this respect.

Other Graphics commands

Both can colour areas of the screen enclosed within plotted llines, and both can put text on the screen. The VICKIT III is rather more sophisticated with these facilities – when it fills in an area on the screen, you can actually find out the area that it has covered and its text plotting is quite versatile.

Both can also determine whether or not a point has been plotted, with the Super Expander additionally being able todetermine the current graphics mode and the contents of the colour registers from within a program.

That's it as far as the Super Expander's graphics are concerned, although it does have several other features which I'll describe later. VICKIT III has some quite powerful additional graphics commands, though.

There is the **DRAW** command, which is really a pseudo-Turtle graphics programming language! It allows a sequence of 'movement' instructions to be built up to draw lines on the screen. These instructions are actually set up as strings and executed using DRAW.

There are also single-letter commands to move the plotting point Up, Down, Left, and Right for a specified distance, leaving a trail as it goes. It can Move to a specified point on the screen (absolute or relative co-ordinates can be used) and it can Change colour.

The sequence can also be Turned through a multiple of 90 degrees or Scaled up or down; and being set up in a string, it is possible to actually call one sequence from within another, therefore allowing whole shape 'subroutines' to be generated. This facility could be very useful for games writing as it allows a limited 'player-graphics' feature, with shapes being set up in strings and called by the DRAW command as required.

Once a graphics screen has been generated, there is a command **PUT** which allows rectangular areas of the screen to be removed and stored in memory. The shape can then be put back again at the same place or at some other place, in the same other place, in the same orientation or inverted horizontally and/or vertically. The memory used for this storage is actually an array variable: and even this can be manipulated. It is stored on a byte-by-byte basis, though, and such manipulations would certainly not be easy.

The DRAW and PUT commands are extremely sophisticated and allow some very clever graphics to be created – once done, there is a **COPY** command which will dump the contents of the screen on to the Vic

Product:	Super Expander	VICKIT III
Supplier:	Commodore	STACK
Price:	£34.95	£21.85
Supplied as:	cartridge (including 3K RAM)	ROM
Located at:	$A000	$A000 or $B000
Initialise with:	Automatic	SYS 40960 or SYS 45056
Disable with:	Not possible	OFF
Setting graphics mode	GRAPHICS mode	GRAPHIC
Selecting colours:	COLOUR b/g, brdr, char, aux	
Background		GROUND colour
Border		BORDER colour
Character/point	REGION colour	SCREEN colour or CLEAR colour
Auxilliary		not available
Setting a point:	POINT 2, x,y [,x1,y1,x2,y2..]	SET (x,y)
Resetting a point:	POINT 0, x,y [,x1,y1,x2,y2..]	RESET (x,y)
Inverting a point:	not available	INVERT (x,y)
Drawing a line:	DRAW c, x,y TO x1,y1 [TO x2,y2..]	LINE (x,y)–(x1,y1) [,k] [can also draw a box]
Drawing a circle:	CIRCLE c,x,y,xr,yr	CIRCLE (x,y),r [,k,h/w]
Drawing an arc:	CIRCLE c,x,y,xr,yr,as,af	CIRCLE (x,y),r,k,h/w,as,af
Fill an area:	PAINT c, x,y	FILL (x,y), k
Text display:	CHAR row, column, "TEXT"	WRITE rule, "TEXT" FONT upr/lwr/own, spacing, addrss
Other commands:	SOUND voice 1,2,3,4, volume KEY number, "STRING" RGR, RCOLR, RDOT (x,y) RJOY, RPEN, RPOT RSND PRINT music string	COPY file, device, upr/lwr, title PUT ary, <>, (x,y)– (x1,y1), rule DRAW string

Parameter Abbreviations:

c	– colour register number	as,af	– arc start and finish points
x,y	– co-ordinates	r	– ordinary circle radius
x1,y1	– second co-ordinates	h/w	– height/width ratio of circle
xr,yr	– x and y radii of circle	k	– indication of Set/Reset/Invert
<>	– transfer direction	rule	– logical rule used to plot point

printer. Unfortunately, the 'aspect ratio' of the printer will be different to the screen, so circles and squares on the screen will come out elongated on the printer.

Incidentally, if you've got a Stack ROM Switchboard, there is a **SWITCH** command too to enable ROMs to be switched in or out.

Beyond Graphics

The VICKIT III is solely a graphics package, but the Super Expander has a range of additional commands available to it.

The **function keys** can all be defined, so that pressing one of them will generate a complete sequence of characters. And the joystick, light pen and paddles can be read into a program using **RJOY, RPEN** and **RPOT** functions.

Sound generation is made easier using one of the two sound commands. First of all, complete chords can be generated using the **SOUND v1,v2,v3,v4,vol** command, where v1-v4 define the pitch of the four oscillators and vol their overall volume. Once this command has been used, the chord stays there until turned off again.

For more 'musical' sound generation, there is a facility which allows musical phrases using single notes to be generated from characters contained within strings in a PRINT statement. Any printed string which starts with CTRL-left arrow is interpreted as a musical phrase and the characters following are 'played'.

The sequence of characters uses the letters A-G for the notes (sharps and flats can also be used); and there are also control characters to set the duration of each note, and to change the tempo, voice, volume and octave. The notes can also be printed on the screen as they are played (as letters of course!) which could be a useful debugging aid.

There are two possible solutions. Including a cursor home character at the start of the music string will ensure that the new line never gets to the bottom of the string. Or every music PRINT command could be ended with a semi-colon to supress the new line, although this does mean that you won't be able to turn the music mode off!

Summary

I find it difficult to choose between these two packages. They are both well thought out and easy to use; but don't think that they'll take *all* the drudgery out of programming – though they will take *some* of it away!

The commands that each provides is shown in the table. I haven't provided a comparative rating on each facility this time simply because they both do their jobs equally well.

Both have reasonable instruction manuals; the VICKIT III manual is rather more difficult to understand, though, probably as a result of the more sophisticated nature of some of the commands.

Range errors (that is, co-ordinates which are out of range) are handled slightly differently in the two packages. They both give ILLEGAL QUANTITY errors for negative co-ordinates; but co-ordinates above the maximum on the VICKIT III are treated as if they were the maximum allowed, with anything over 255 generating an error. The Super Expander accepts values over the upper limit, but ignores them.

If you already use the Stack VICKIT I, then I would suggest going for the VICKIT III; if you have the Commodore Programmers Aid, go for the Commodore package . . . but be warned that the two packages are not fully compatible. For instance, problems can occur if you try to find Super Expander keywords using the Programmers Aid!

Overall, I would suggest the Commodore Super Expander is a slightly better buy with those extra commands available. On the other hand, the VICKIT III could be worth considering if you don't want these extras but would like arather more sophisticated graphics package.

Catalogue of Sounds

by Pauli Terho

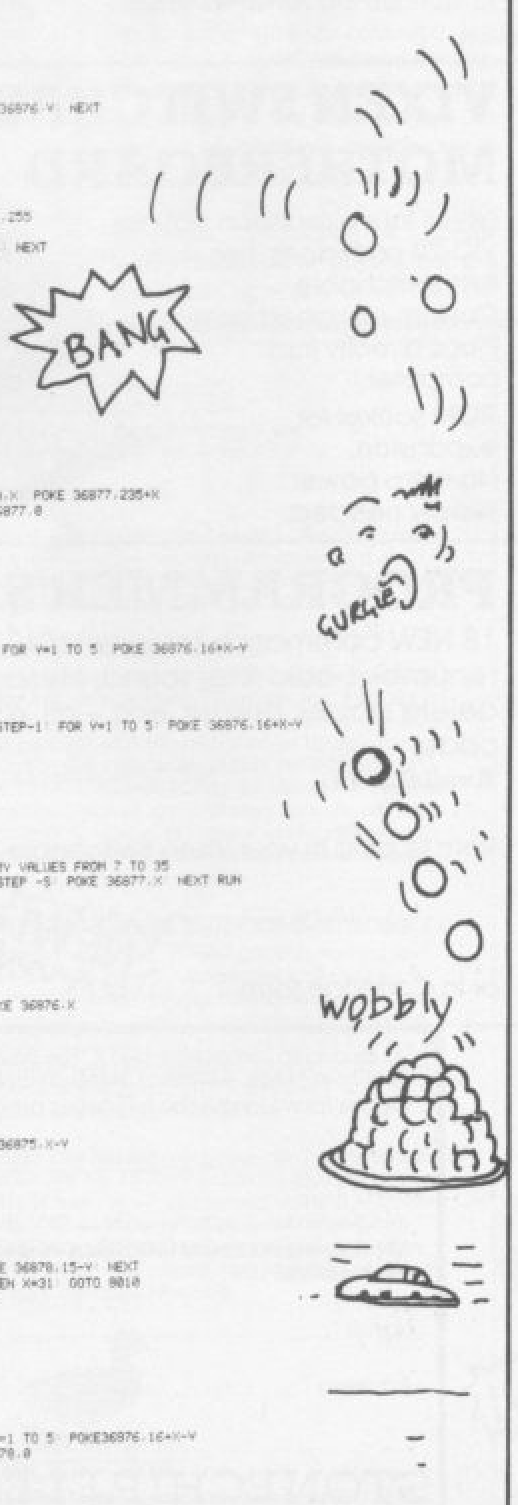
This collection of sound-generating routines took over six months to get from Pauli to us (Sweden isn't that far away) and arrived mangled in a Post Office bag. Deciphering the bits we ended up with this collection of sound routines for the unexpanded Vic.

We tweaked one or two of them and ran them together into one program just so that we didn't lose any; pick out the routine you want.

We're interested in hearing any other interesting sounds, so keep them coming folks!




```
1000 REM ========================
1001 REM === REVERBERATIONS ===
1002 REM ========================
1003 REM
1010 REM ONE TYPE ...
1020 FOR X=1 TO 35: POKE36878,15
1030 FOR Y=254 TO 255-Z STEP -9: POKE 36876,Y: NEXT
1040 Z=Z+1: NEXT: POKE 36876,0
1097 END
1098 REM
1099 REM
1100 REM ... ANOTHER TYPE
1101 REM ... ANOTHER TYPE
1110 FOR X=10 TO 0 STEP -1: POKE 36876,255
1120 FOR Y=1 TO 40: NEXT: POKE 36876,0
1130 FOR Y=1 TO 40: NEXT: POKE 36878,X: NEXT
1197 END
1198 REM
1199 REM
2000 REM =============
2001 REM === SHOT ===
2002 REM =============
2010 POKE 36878,10
2020 FOR X=1 TO 20: POKE36874,251-X
2030 NEXT: POKE36874,0
2197 END
2198 REM
2199 REM
3000 REM =============
3001 REM === WIND ===
3002 REM =============
3010 FOR X=15 TO 0 STEP -1: POKE 36878,X: POKE 36877,235+X
3020 FOR Y=1 TO 120: NEXT Y,X: POKE 36877,0
3197 END
3198 REM
3199 REM
4000 REM ===============
4001 REM === GURGLE ===
4002 REM ===============
4003 REM
4010 REM GOING UP ...
4020 POKE 36878,10: FOR X=219 TO 239: FOR Y=1 TO 5: POKE 36876,16+X-Y
4030 NEXT: NEXT
4040 REM ... GOING DOWN
4050 REM
4060 REM ... GOING DOWN
4070 POKE 36878,10: FOR X=239 TO 219 STEP-1: FOR Y=1 TO 5: POKE 36876,16+X-Y
4080 NEXT: NEXT
4090 POKE 36876,0
4197 END
4198 REM
4199 REM
5000 REM ====================
5001 REM === HELICOPTER ===
5002 REM ====================
5010 S=10:REM ADJUSTS ROTOR SPEED - TRY VALUES FROM 7 TO 35
5020 POKE 36878,15: FOR X=255 TO 128 STEP -S: POKE 36877,X: NEXT:RUN
5197 END
5198 REM
5199 REM
6000 REM ===============
6001 REM === BOUNCE ===
6002 REM ===============
6010 FOR Y=1 TO 15: POKE 36878,15-Y
6020 FOR X=255-Y TO 220-Y STEP -Y: POKE 36876,X
6030 NEXT: NEXT: POKE 36876,0
6197 END
6198 REM
6199 REM
7000 REM ===============
7001 REM === WOBBLY ===
7002 REM ===============
7010 FOR Y=1 TO 15: POKE 36878,15-Y
7020 FOR X=239-Y TO 254 STEP 3: POKE 36875,X-Y
7030 NEXT: NEXT: POKE 36875,0
7197 END
7198 REM
7199 REM
8000 REM ==============
8001 REM === ALIEN ===
8002 REM ==============
8010 POKE 36874,X: FOR Y=0 TO 15: POKE 36878,15-Y: NEXT
8020 FOR T=1 TO 40: NEXT: IF X=255 THEN X=31: GOTO 8010
8030 REM INCREASE T TO INCREASE DRAMA
8040 X=255: GOTO 8010
8197 END
8198 REM
8199 REM
9000 REM ================
9001 REM === FLIPPER ===
9002 REM ================
9010 POKE 36878,10
9020 FOR X=239 TO 210: STEP -1: FOR Y=1 TO 5: POKE36876,16+X-Y
9030 NEXT: NEXT: POKE36876,0: POKE36878,0
9197 END
```



The Over Command

by Leonard Morrow

Not so much a routine, more a revelation. Did you know that there's a hidden command in the Super Expander? Leonard Morrow found it: it's OVER, and it allows you to draw, paint or circle over a coloured line in the background colour.

You use it as a suffix — DRAWOVER, PAINTOVER or CIRCLEOVER. What it does is to remove a graphic line from the screen without removing any other lines; so it's a kind of controlled version of SCNCLR, which of course removes everything.

This short program demonstrates the availability of OVER as used with DRAW.




```
1 REM *** DRAWOVER DEMO ***
2 REM *** FOR SUPER EXPANDER ***
3 REM *** BY LEONARD MORROW ***
10 GRAPHIC2: COLOR 0,0,1,1
20 CHAR 17,2,"DEMONSTRATING THE"
29 CHAR 19,2,"COMMAND 'DRAWOVER'"
30 X=0: Y=100: X1=0: Y1=600
35 IF X>1020 THEN X=0: X1=0
40 DRAW2,X,YTOX1,Y1
50 X=X+6: X1=X1+6
60 DRAWOVER2,X-6,YTOX1-6,Y1
70 GOTO35
```

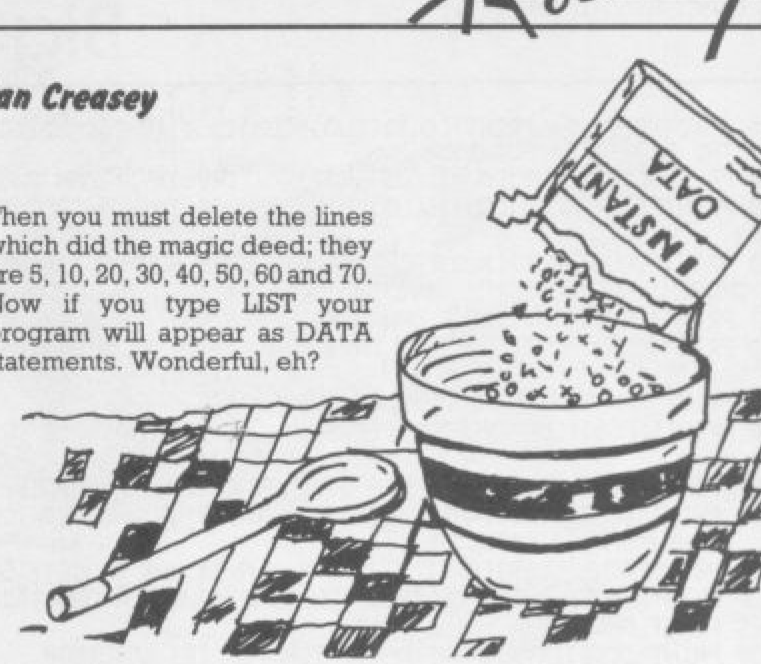
On error go to

by Andy Pearce

```
1 GOTO100
2 SUBROUTINE TO
3 SIMULATE THE
4 ON ERROR GOTO
5 INSTRUCTION
6 FOUND ON SOME
7 COMPUTERS
8 ADAPTED FROM
9 A SUBROUTINE
10 FOR THE PET
11 GIVEN IN
12 'PRACTICAL
13 COMPUTING'
14 OCTOBER 1982
15 ORIGINAL WRITTEN
16 BY
17 PAUL BRADSHAW
18 SUNDERLAND
19 THIS VERSION
20 (FOR VIC 20)
21 BY
22 ANDY PEARCE
23 DERBY
100 REM- WHERE YOU WOULD TYPE ONERR GOTO 1000
101 REM USE THE NEXT LINE INSTEAD
102 L=1000 GOSUB60000
103 REM PUT YOUR PROGRAM HERE
195 REM EXAMPLE ACTION
199 X=100 Y=100
200 FOR2
201 READ2,X,Y
202 Y=Y-1
203 GOTO201
210 REM
999 REM PUT YOUR ERROR HANDLING ROUTINE HERE
1000 RUN
1001 REM
59999 END
60000 A$="GOTO"+STR$(L)+CHR$(13)+CHR$(13)
60010 FORJ=1TOLEN(A$)
60020 POKE631+J,ASC(MID$(A$,J,1))
60030 NEXTJ
60035 REM THIS HAS POKED 'GOTO 1000' AND A CARRIAGE RETURN INTO THE KEYBOARD BUF
FER
60040 REM SO THAT WHEN THE PROGRAM TRIES TO RETURN TO COMMAND MODE IT IS INSTEAD
60045 REM DIRECTED TO YOUR ERROR HANDLING ROUTINE AT LINE 1000 (OR WHEREVER)
60050 POKE198,J-1
60055 REM NUMBER OF CHARACTERS LEFT IN BUFFER
60060 REM YOU SHOULD POKE 198,0 BEFORE ANY INPUT OR GET STATEMENTS AND JUST BEFO
RE
60065 REM THE END OF YOUR PROGRAM
60070 RETURN
```

Instant data

by Ian Creasey



One of our more prolific regular correspondents, Ian Creasey from Stockport, has sent us a whole wadge of stuff over the past few months. Here's a clever little program that helpfully creates lines of Basic with your machine code as DATA in it — as Ian remarks, it's a real drag having to write loads of DATA statements in a Basic loader when you've written a machine-code program.

All you have to do is type the start line number, the start of the machine code, and the end of the code: separate those entries with commas. Presto! New lines of DATA will be created.

When it's running you may get a series of depressing screen messages (like OUT OF DATA, RUN, READY, GOTO) — but you can ignore them all. The one you're looking for is 'OK', which means the program is finished.

Then you must delete the lines which did the magic deed; they are 5, 10, 20, 30, 40, 50, 60 and 70. Now if you type LIST your program will appear as DATA statements. Wonderful, eh?

```
5 REM***M-C TO DATA***BY IAN CREASEY***
10 INPUTSL,S,E:POKE0,SL/10:POKE2,S/256:POKE1,S-PEEK(2)*256:POKE194,E/256:POKE193
,E-PEEK(194)*256
20 I$="GOTO50"+CHR$(13):FORI=0TO6:POKE631+I,ASC(MID$(I$,I+1,1)):NEXT:POKE198,7
30 FORI=PEEK(1)+PEEK(2)*256TOPEEK(193)+PEEK(194)*256:READJ:IFPEEK(I)<>JTHENPOKE1
98,0:STOP
40 NEXT:PRINT"OK":POKE198,0:END
50 PRINTPEEK(0)*10"DATA";:POKE0,PEEK(0)+1:J=I+15:K=PEEK(193)+PEEK(194)*256:IFJ>K
THENJ=K
60 FORI=ITOJ:I$=STR$(PEEK(I)):PRINTRIGHT$(I$,LEN(I$)-1)",";:NEXT:PRINT"[CUL] ":P
RINT"RUN20";
70 POKE631,145:POKE632,145:POKE633,145:POKE634,13:POKE635,13:POKE198,5
```

Random Array

This routine from Dr. D. Barton of Hampstead fills any array with different random numbers.

If you are dealing with a pack of cards, you obviously don't want the Ace of Spades to be dealt twice (although funny things like that do happen when our Production Editor is playing poker). The program has uses three variables, X, Y and Z, which could be replaced by constants in a particular application.

These variables have the following uses:

X: number of possible numbers
Y: lowest possible number
Z: number of random numbers required.

by D Barton

The values are integers and are held in an array Q, which should have been dimensioned earlier in the program.

Note that Z must be =X, and that the highest possible number is X+Y. So here, for your delectation, is the routine:

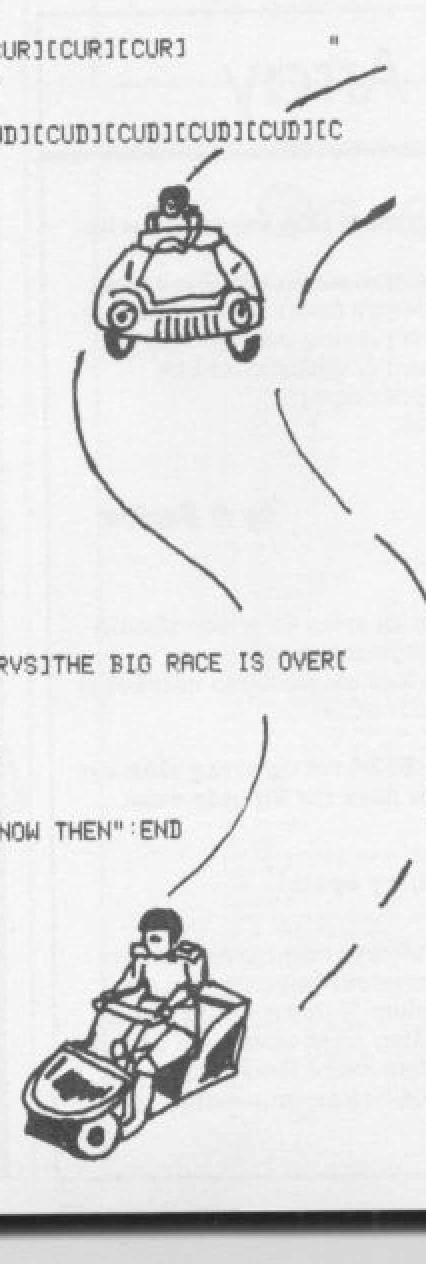
```
100 FOR J=1 TO Z
105 Q(J)= INT (RND (1) * X) + Y : C=0 :REM set up array element
110 FOR K=1 TO J : REM check number does not already exist
115 IF Q(J) = Q(J-K) THEN C=1
120 NEXT K
125 IF C=1 THEN 105 : REM yes it does, try again!
130 NEXT J
```

There is a slight problem in that line 115 always compares element Q(0). This presumably contains 0 (as it has never been set up), so none of the other elements can have 0 either. If this *is* going to be a difficulty you can alter the routine by adding an IF statement before line 110 to avoid the test when J=1 and then make the K-loop run from 1 to J-1. A neater way is just to set Q(0) to an impossible value, say -1, before the start of the routine.

Victuals

Big race

```
10 POKE36879,124:PRINT"[CLR][CUD][CUD][CUD][CUD][CUD][CUD][CUD][CUD][RED]     [R
VS]THE BIG RACE":GOSUB550
20 PRINT"[CUD][CUD][CUD][CUD][WHT]    [RVS]BY  JOHN TULLY":GOSUB550
30 PRINT"[CUD][CUD][CUD][CUD]THIS GAME IS FOR UP TO[CUD]     SIX  PLAYERS":GOSUB
550
40 PRINT"[CUD][CUD][BLU]EACH PLAYERS NAME MUST[CUD]  BE ENTERED AGAINST  [CUD]TH
E CHOSEN LANE NUMBER":GOSUB550
50 PRINT"[CUD][RED]  FOR UNWANTED LANES          [CUD]HIT [RVS]RETURN":GOSUB550
60 PRINT"[CUD][CUD][BLK]HIT [RVS]SPACE[RVO] TO START THE[CUD]RANDOM NUMBER SEQUE
NCE"
70 PRINT" IN THE 'LANE ADVANCE'":GOSUB550
80 PRINT"[CUD][CUD][PUR]HIT [RVS]SPACE[RVO] AGAIN TO TRY[CUD]  TO STOP THE NUMBE
R"
90 PRINT"[CUD] ON [RVS]YOUR[RVO] LANE NUMBER":GOSUB550
100 PRINT"[CUD][CUD][WHT]THE NUMBER YOU STOP ON[CUD]WILL ADVANCE THAT LANE[CUD]
    ONE  SPACE":GOSUB550
110 PRINT"[CUD][CUD][RVS][BLU]EACH PLAYER HAS 3 GOES":GOSUB550
120 PRINT"[CUD][CUD][RVS][PUR][CUR]THE COMPUTER CHOOSES[CUR][CUR][CUR][CUD]WHO G
OES FIRST AND"
130 PRINT"[CUD][RVS]SHOWS WHOSE TURN IT IS":GOSUB550
140 PRINT"[CUD][CUD][RED]THE FIRST TO REACH THE   [CUD]BOTTOM LINE WINS":GOSUB55
0
150 PRINT"[CUD][CUD][BLU] IF NOBODY HAS CHOSEN    [CUD]THE WINNING LANE"
160 PRINT"[CUD]THEN THE COMPUTER WINS":GOSUB550
170 POKE36879,232:PRINT"[CLR][RVS][PUR]  ** THE BIG RACE **  [RVO] [RED]  LANE N
UMBER 123456"
180 FORN=0TO440STEP22:POKE7702+N,160:POKE38422+N,1:POKE7723+N,160:POKE38443+N,1
190 POKE7716+N,160:POKE38436+N,1:NEXT:PRINT"[HOM][CUD][CUD][WHT][RVS]_________ST
ART_[RVO][RED]♠[CYN]●[BLU]♣[PUR]♦[YEL]*[BLK]●[RVS][WHT]_"
200 FORN=0TO21:POKE8164+N,170:POKE38884+N,4:NEXT
210 PRINT"[HOM][CUD][CUD][CUD][CUD][CUR][RED]NEXT PLAYER":PRINT"[CUD][CUR]IS":PR
INT"[CUD][CUD][CUR][WHT]ADVANCE":PRINT"[CUD][CUR]LANE"
220 PRINT"[CUD][CUD][CUR][YEL][RVS]LANE":INPUT"[CUR]1[PUR]";P1$:INPUT"[YEL][CUR]
2[PUR]";P2$:INPUT"[YEL][CUR]3[PUR]";P3$
230 INPUT"[YEL][CUR]4[PUR]";P4$:INPUT"[YEL][CUR]5[PUR]";P5$:INPUT"[YEL][CUR]6[PU
R]";P6$
240 P=INT(RND(1)*6)+1
250 RESTORE
260 A=7739:B=7740:C=7741:D=7742:E=7743:F=7744:K=0:J=0
270 K=K+1:IFK=4THENK=1:P=P+1
280 GOSUB560:PRINT"[HOM][CUD][CUD][CUD][CUD][CUD][CUD][CUR][CUR][CUR]          "
:PRINT"[CUP][BLK][RVS][CUR][CUR][CUR][CUR]";LEFT$(P$,10)
290 GETA$:IFA$<>" "THEN290
300 S=INT(RND(1)*6)+1:PRINT"[HOM][CUD][CUD][CUD][CUD][CUD][CUD][CUD][CUD][CUD][C
UD][CUD][CUR][CUR][CUR][CUR][CUR]";S
310 FORT=0TO400:NEXT
320 GETB$:IFB$<>" "THEN300
330 IFS=1THENX=A:Y=2:Z=65:A=A+22
340 IFS=2THENX=B:Y=3:Z=81:B=B+22
350 IFS=3THENX=C:Y=6:Z=88:C=C+22
360 IFS=4THENX=D:Y=4:Z=90:D=D+22
370 IFS=5THENX=E:Y=7:Z=42:E=E+22
380 IFS=6THENX=F:Y=0:Z=83:F=F+22
390 POKEX,32:X=X+22:POKEX,Z:POKEX+30720,Y
400 IFX>8164THEN420
410 GOTO270
420 POKE36878,15:READQ,R
430 DATA231,3,225,3,215,6,0,2,195,2,215,6,-1,-1
440 IFQ=-1THENPOKE36878,0:POKE36875,0:GOTO460
450 POKE36875,Q:FORT=1TO300*R:NEXT:GOTO420
460 J=1:P=S:GOTO570
470 IFP$=""THENP$="VIC       "
480 POKE36879,25:PRINT"[CLR][CUD][CUD][CUD][CUD] [RED][RVS]THE BIG RACE IS OVER[
CUR][CUR][CUR][CUR][CUR][CUD][CUD][GRN]THE WINNER IS"
490 PRINT"[CUD][CUD]       [RVS][PUR]";LEFT$(P$,10)
500 PRINT"[CUD][CUD] [RVS][CYN]WELL DONE ";LEFT$(P$,10)
510 PRINT"[CUD][CUD]  [RVS][BLU]RACE AGAIN ? (Y/N)"
520 GETR$:IFR$="Y"THEN170
530 IFR$="N"THENPRINT"[CUD][CUD] [RVS][YEL]BYE BYE FOR NOW THEN":END
540 GOTO520
550 FORT=0TO2000:NEXT:RETURN
560 IFP>6THENP=P-6
570 IFP=1THENP$=P1$
580 IFP=2THENP$=P2$
590 IFP=3THENP$=P3$
600 IFP=4THENP$=P4$
610 IFP=5THENP$=P5$
620 IFP=6THENP$=P6$
630 IFJ=1THEN470
640 IFP$=""THENP=P+1:GOTO560
650 RETURN
```



by J A Tully

This is a game for up to six players, in which each player chooses a lane and tries to advance the symbol in that lane to the bottom of the screen.

The computer asks for the players' names to be entered in sequence against the chosen lane numbers. If there are less than six players then hit return for the lanes not chosen (these can be any of the six lanes). The computer then decides at random which player goes first, and displays the name on the screen. Thereafter, the players go in the order of lane numbers, the sequence being repeated until the race is won.

To play the game, the first player hits the space bar. This starts a sequence of random numbers which are displayed consecutively on the screen. Hitting the space bar a second time halts the display and the symbol in the lane corresponding to the number showing is advanced one space.

Subsequently hitting the space bar alternately starts and stops the display. Each player has three consecutive goes after which the name of the next player will be displayed on the screen.

The game ends when one symbol reaches the bottom of the screen. This will result in a fanfare and the winner's name being displayed. If the winning symbol was not one of those selected by the players, then the computer will be declared the winner.

List of variables:

- A,B,C,D,E,F are the positions of the various symbols in the six lanes
- Y and Z are the shape and colour of the symbol to be moved
- S and X determine which lane advances
- P controls the next player display

Program Sections

- Lines 280-320 print the background display
- Lines 330-340 ask for the players' names
- Line 350 select which player goes first
- Lines 370-520 control the motion of the symbols
- Line 410 is the random lane number generator
- Lines 530-560 play fanfare
- Line 600 controls speed of instructions at beginning of program

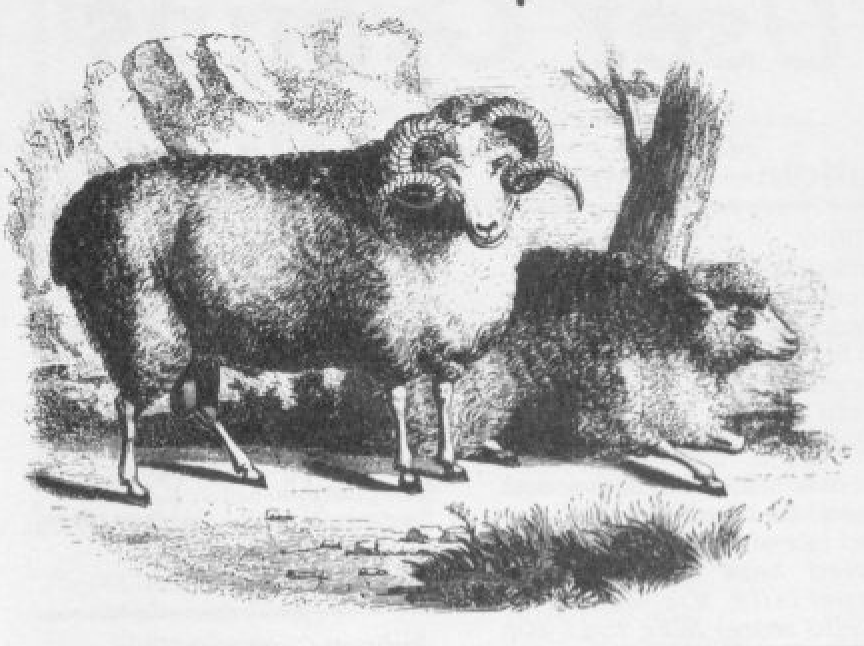
Sheepdog

by James Newall

It's 'One Man and a Dog' time: you have to move the sheepdog to round up the sheep (you pick how many sheep you want to work with) and get them into the pen. The program's verisimilitude is illustrated by the built-in stupidity of the sheep and their tendency to bolt if your dog gets too close; when the sheep are all penned, move the dog into the entrance to stop them escaping (and to stop the game too).

Our only criticism: it's possible to park the dog right next to a sheep that simply won't budge at all — not very realistic. But generally a good game.

```
1 POKE36879,26
2 POKE36878,15
5 DIMX(6),Y(6)
10 PRINT"[CLR]"
20 FORI=1TO22:PRINT"[GRN]■";:NEXTI
30 PRINT"[CUD][RVS][RED]*** SHEEP DOG ********"
40 PRINT"[CUD][CUD] TRY TO ROUND UP THE"
50 PRINT"[CUD][CUR][CUR]SHEEP INTO THE PEN"
60 PRINT"[CUD][CUR]WITH YOUR SHEEPDOG."
70 PRINT"[CUD][CUD][CUR][CUR][BLK][RVS]CONTROLS"
80 PRINT"[CUD][CUR][BLU]T=UP   V=DOWN "
90 PRINT"[CUD][CUR]F=LEFT  H=RIGHT"
100 PRINT"[CUD]  PRESS ANY KEY"
110 GETA$:IFA$=""THENGOTO110
115 PRINT"[CLR]"
120 GOSUB5000
130 PRINT"[RVS]HOW MANY SHEEP ([RVO]";"B";"[RVS])"
135 PRINT"[CUD][RVS]UPTO 6"
140 GETA$:IFA$=""THEN140
141 CF=VAL(A$):IFCF>6THENPRINT"[RVS][RED]NO MORE THAN 6":GOTO140
150 PRINT"[CUD][CUD][RED][RVS]PRESS ANY KEY[CUD][CUD][CUL][CUL][CUL][CUL]TO STAR
T"
160 GETA$:IFA$=""THEN160
166 TI$="000000"
170 E$="[CUR][CUR][CUR][CUR][CUR][CUR][CUR][CUR][CUR][CUR][CUR][CUR][CUR][CUR][C
UR][CUR][CUR][CUR][CUR][CUR][CUR]"
180 P$="[CUD][CUD][CUD][CUD][CUD][CUD][CUD][CUD][CUD][CUD][CUD][CUD][CUD][CUD][C
UD][CUD][CUD][CUD][CUD][CUD][CUD]"
190 PRINT"[CLR]";LEFT$(E$,15);LEFT$(P$,10);"[RVS] [RVO][WHT] [RED][RVS] "
200 PRINTTAB(15);"[RVS] [RVO][WHT]   [RED][RVS] [RVO]":PRINTTAB(15);"[RVS] [RVO]
[WHT]   [RED][RVS] [RVO]":PRINTTAB(15);"[RVS]     "
205 PRINT"[HOM]";"[RVS]*****SHEEP  DOG*******[RVO]"
210 PRINT"[HOM][CUD]";:FORI=1TO22:PRINT"[GRN][RVS]■";:NEXTI
220 FORI=1TO18:PRINT"[HOM]";LEFT$(P$,I);"[GRN][RVS]■";LEFT$(P$,20);"[GRN][RVS]■"
;:NEXTI
230 PRINT"[HOM]";LEFT$(P$,14):FORI=1TO66:PRINT"[GRN][RVS]■";:NEXTI
235 PRINT"[HOM]";LEFT$(P$,20);"[BLU]TIME:"
240 FORI=1TOCF
250 X=INT(RND(1)*21)+1:Y=INT(RND(1)*15)+1
260 IFPEEK(7680+X+22*Y)<>32THEN250
265 IFI=1THENPRINT"[HOM]";LEFT$(E$,X);LEFT$(P$,Y);" "
270 XV(I)=X:YV(I)=Y:PRINT"[HOM]";LEFT$(E$,XV(I));LEFT$(P$,YV(I));"[CYN]B":NEXTI
280 X=INT(RND(1)*21)+1:Y=INT(RND(1)*15)+1
290 IFPEEK(7680+X+22*Y)<>32THEN280
300 XD=X:YD=Y:PRINT"[HOM]";LEFT$(E$,XD);LEFT$(P$,YD);"[BLK]A"
310 FORI=1TO3:GOSUB3000
320 X=XD:Y=YD
330 IFG$="T"THENY=YD-1
340 IFG$="V"THENY=YD+1
350 IFG$="F"THENX=XD-1
360 IFG$="H"THENX=XD+1
370 IFPEEK(7680+X+22*Y)<>32THENX=XD:Y=YD
380 PRINT"[HOM]";LEFT$(E$,XD);LEFT$(P$,YD);"[WHT] "
390 XD=X:YD=Y:PRINT"[HOM]";LEFT$(E$,XD);LEFT$(P$,YD);"[BLK]A"
391 POKE36874,189:FORSO=1TO10:NEXTSO:POKE36874,0
392 GOSUB7000
393 NEXTI
400 FORI=1TOCF
410 X=XV(I):Y=YV(I)
420 IFABS(XV(I)-XD)>3ORABS(YV(I)-YD)>3THEN490
430 IFABS(XV(I)-XD)<2ANDABS(YV(I)-YD)<2THEN490
440 IFXV(I)>XDTHENX=XV(I)+1
450 IFXV(I)<XDTHENX=XV(I)-1
460 IFYV(I)>YDTHENY=YV(I)+1
470 IFYV(I)<YDTHENY=YV(I)-1
480 GOTO500
490 X=XV(I)+1-INT(3*RND(1)):Y=YV(I)+1-INT(3*RND(1))
500 IFPEEK(7680+X+22*Y)<>32THENX=XV(I):Y=YV(I)
510 PRINT"[HOM]";LEFT$(E$,XV(I));LEFT$(P$,YV(I));"[WHT] "
520 XV(I)=X:YV(I)=Y:PRINT"[HOM]";LEFT$(E$,XV(I));LEFT$(P$,YV(I));"[CYN]B"
525 POKE36874,160:FORSO=1TO2:NEXTSO:POKE36874,0
526 GOSUB7000
530 NEXTI
550 GOTO310
3000 GETG$
3050 PRINT"[HOM]";LEFT$(E$,6);LEFT$(P$,20);"[RED][RVS]";TI$;"[RVO]"
3060 IFG$=""THEN3000
3100 RETURN
5000 DATA0,7,71,120,120,120,72,72
5005 DATA0,0,125,255,249,121,72,72
5010 FORI=7176TO7183
5020 READS:POKEI,S:NEXTI
5030 FORI=7184TO7191
5040 READD:POKEI,D:NEXTI
5045 FORI=7192TO7200
5046 POKEI,0
5047 NEXTI
5050 POKE36869,255
5060 RETURN
7000 GH=0:FORU=1TO3:IFPEEK(7937+U)=2THENGH=GH+1
7010 IFPEEK(7959+U)=2THENGH=GH+1
7020 IFGH=CFTHEN7040
7025 NEXTU
7026 GH=0
7030 RETURN
7040 IFPEEK(7917)=1THENDE$=TI$:GOTO8000
7045 GH=0
7050 RETURN
8000 FORI=250TO128STEP-1
8001 POKE36875,I:FORT=1TO5:NEXTT:NEXTI
8002 FORT=1TO30:NEXTT:FORI=128TO250:POKE36875,I:FORT=1TO5:NEXTT:NEXTI:POKE36875,
0
8003 POKE36879,30
8004 FORV=30TO155:POKE36865,V:FORJ=1TO5:NEXTJ:NEXTV
8005 PRINT"[CLR]"
8006 POKE36879,27:POKE36869,240
8007 PRINT"[CUD][CUD][CUD][RED][RVS]END  OF GAME[RVO]":PRINT"[CUD][BLK]YOU HAVE
ROUNDED UP"
8008 PRINT"[CUD]";CF;"SHEEP IN A TIME"
8009 PRINT"[CUD]OF";DE$;".";"IF YOU WANT":PRINT"[CUD]ANOTHER GO ENTER 1":PRINT"[
CUD]IF NOT ENTER 2"
8010 FORV=155TO30STEP-1:POKE36865,V:POKE36876,230:FORT=1TO3:NEXTT:POKE36876,0:NE
XTV
8013 INPUTFR:IFFR=1THENRESTORE:RUN
8014 IFFR=2THENPRINT"[BLU]GOODBYE YOU SCUM":FORT=1TO100:NEXTT:PRINT"[CLR]"
```



Poster print

If you have a Vic with at least 8K expansion and a printer, you might like this poster designer. It only gives capitals and graphics, but it's quite fun.

The screen is formatted with a 15x10 grid and an asterisk cursor which can be moved around to construct poster text. When the poster is complete, pressing the back arrow key (←) removes the cursor and the program gets on with preparing the poster for print — this is rather slow, so a progress report is given — and eventually the poster is output to the printer. The option for immediately reprinting is then given.

Lines 160-200 format the screen, 210-220 enable entry of poster and screen characters, and the GOSUB 520 positions the cursor within the grid.

Lines 340-380 scans the grid for preparation. That GOSUB 570 checks the character ROM for bit values of the character and forms a print string for each line in L$(15,77).

Lines 390-420 print the poster. The GOSUB 500 prints a blank line (flagged by BL(15) to speed up blank lines.

Lines 430-490 prompt for reprint or a new poster.

by Roger Peacock

```
100 REM  *** POSTERPRINT ***
110 REM
120 REM  BY ROGER PEACOCK
130 REM
140 REM****NEEDS 8K+ AND PRINTER*****
150 REM
160 DIML$(15,7),BL(15):SM=4096:PRINTCHR$(142);CHR$(8)
170 PRINT"[CLR][BLK][RVS]POSTERPRINT
180 FORJ=1TO15:PRINT"[RVO]          [RVS][GRN]
190 NEXT:PRINT"
200 PRINT"[HOM][CUD][CUD][CUD][CUD][CUD][CUD][CUR][CUR][CUR][CUR][CUR][CUR][CUR]
[CUR][CUR][CUR][CUR][CUR][CUR][CUR][CUR][BLK]ENTER";:PRINT"[CUD][CUL][CUL][CUL][
CUL]AND[CUD][CUL][CUL][CUL][CUL]PRESS[CUD][CUL][CUL][CUL][CUL][CUL]'←'TO[CUD][CU
L][CUL][CUL][CUL][CUL]PRINT"
210 R=2:C=0:GOSUB520:PRINT"[BLK]";
220 GOSUB520:PRINT"*";
230 GETA$:IFA$=""THEN230
240 PRINT"[CUL] [CUL]";
250 IFA$="←"THEN340
260 CD=ASC(A$):IFCD=13ANDR<18THENC=0:R=R+1:GOSUB520:GOTO230
270 IFCD=13THEN220
280 IFCD=17THENR=R+1:GOTO220
290 IFCD=29THENC=C+1:GOTO220
300 IFCD=145THENR=R-1:GOTO220
310 IFCD=157ORCD=20THENC=C-1:GOTO220
320 PRINT"[BLK]";A$;:C=C+1:IFC>9THENC=0:R=R+1
330 GOTO220
340 PRINT"[HOM][CUD][RVS]PREPARING":LINE=0
350 FORA=SM+44TOSM+352STEP22:LINE=LINE+1:PRINT"[HOM][CUR][CUR][CUR][CUR][CUR][CU
R][CUR][CUR][CUR][CUR][CUR][CUR]LINE";LINE
360 FORB=ATOA+9
370 N=PEEK(B):GOSUB570
380 NEXT:NEXT
390 PRINT"[HOM][CUD][RVS] PRINTING "
400 OPEN4,4:FORLINE=1TO15:IFBL(LINE)=0THENGOSUB500:GOTO420
410 FORA=0TO7:PRINT#4,L$(LINE,A);:PRINT#4,CHR$(8):PRINT#4,CHR$(15);:NEXT
420 NEXT:CLOSE4
430 PRINT"[HOM][CUD]                  "
440 C=12:R=14:GOSUB560
450 PRINT"[BLK][RVS]R[RVO]EPRINT":R=16:GOSUB560:PRINT"[BLK][RVS]N[RVO]EW TEXT":P
RINT"[CUD][CUR][CUR]PLEASE SELECT*";
460 GETS$:IFS$=""THEN460
470 IFS$="R"THEN400
480 IFS$="N"THENRUN
490 PRINT"[CLR]";:END
500 REM*BLANK*LINE***
510 FORJ=1TO4:PRINT#4:NEXT:RETURN
520 IFR<2THENR=2:RETURN
530 IFC<0THENC=0:RETURN
540 IFR>16THENR=16:RETURN
550 IFC>9THENC=9:RETURN
560 POKE781,R:POKE782,C:POKE783,0:SYS65520:RETURN
570 REM FORM STRING
580 IFN<>32THEN600
590 FORJ=0TO7:L$(LINE,J)=L$(LINE,J)+"        ":NEXT:RETURN
600 FORJ=0TO7:L=PEEK(32768+(N*8)+J):CC=128:FORK=1TO8
610 IFL-CC>=0THENL$(LINE,J)=L$(LINE,J)+"■":L=L-CC:GOTO630
620 L$(LINE,J)=L$(LINE,J)+" "
630 CC=CC/2:NEXT:NEXT:BL(LINE)=1:RETURN
```

Users

User Groups

We've been collecting and updating our information on users' clubs and groups that Victims might want to join. Here's the list so far — and clubs not listed are invited to contact us with information.

Britain—national

ICPUG
(Independent Commodore Products User Group)
Membership Secretary
30 Brancaster Road
Newbury Park
Ilford
Essex IG2 7EP

...established and active, with good bi-monthly newsletter and lots of local groups (some listed here if they seem especially Vic-oriented). ICPUG shares Mike Todd with us for Vic matters. Annual sub £7.50.

London

Association of London Computer Clubs
13 Compton Road
London N1 2PA

...an umbrella organisation for two dozen London-area organisations; ALCC organises the London Computer Fair among other things.

North London Hobby Computer Club
c/o Dept of Electronic & Computer Engineering
Polytechnic of North London
Holloway Road London N7 8DB

...£25 pa for adults, £5 for unemployed and Poly students. Highly organised, 250 members, lots of subsections including one for Vic. They also run the Association of London Computer Clubs.

Vic 20 User Group
c/o Jim Chambers
Dept of Psychology
University College
26 Bedford Way
London WC1. *30p per meeting*

Berks

ICPUG Slough
c/o Brian Jones
Slough College of Higher Education
Wellington Street
Slough

...£5 pa. Includes Vics.

Bucks

Richard Onion
19 Frogmoor Close
Hughenden Valley
High Wycombe

...was trying to set up a local Vic group.

Derby

ICPUG Derby
c/o Ray Davies
105 Normanton Road
Derby DE1 2GG

...£3 pa. Includes Vics.

Derby Microcomputer Society
c/o M. Riordan
Littleover Church Hall
Shepherd Street
Littleover. *£5 pa. Includes Vics.*

Dorset

Bridport Computer Club
c/o M. J. Higgins
BBC Transmitting Station
Rampisham Down
Maiden Newton
Dorchester. *Started in January.*

Bournemouth Computer Club
c/o Peter Dibbs
54 Runnymede Avenue
Bournemouth BH11 9SE

...£5 pa, 100 members. Includes Vics.

Essex

Basildon Computer Club
c/o Peter Silver
24 Havengore
Pitsea *...£5 pa. Includes Vics.*

Brentwood Microcomputer Club
c/o Allan Holland
212 Woodland Avenue
Hutton
Brentwood CM13 1DA

...Includes a section devoted to Commodore users, mostly Vics.

ICPUG Essex
c/o Tony Surridge
97 Shelley Road
Chelmsford

Gloucester

ICPUG West of England
c/o Janet Rich
Rose Cottage 20 Old Court
Springhill
Glos GL11 5PF

Hampshire

ICPUG Hants
c/o Ron Geere
109 York Road
Farnborough

Southampton ACC
c/o Paul Blitz
Gardenways
Chilworth Tower
Chilworth
Southampton SO1 7JH

...Local branch of Amateur Computer Club. 100 members, £5 pa: includes Vics.

Herts

ICPUG North Herts
c/o Brian Grainger
73 Minehead Way
Stevenage SG1 2HZ

...£7.50 pa. Includes Vics.

CRS Home Computer Club
c/o Robert Crutchfield
2 Durham Road
Stevenage SG1 4HS

...50p per meeting. Includes Vics.

Humberside

Grimsby Computer Club
c/o J. Lee
29 Park View
Cleethorpes DN35 7TG

...£8 pa. Includes Vics.

Scunthorpe Computer Club
c/o D. Needham
37 High Lees Road
Scunthorpe DN17 2QA

...£5 pa. Includes Vics.

Kent

Gravesend Computer Club
c/o The Extra Tuition Centre,
39 The Terrace, Gravesend.
Kent DA12 2BA

...£6 pa (£3 for unwaged). About fifty members with a variety of machines including Vic.

Orpington Computer Club
c/o Roger Pyatt
23 Arundel Drive Orpington

...Includes Vics — and CB/RTTY, if that's your bag.

North Kent ACC
c/o Iain House
28 Canadian Avenue
Catford London SE6 3AS

...Local branch of Amateur Computer Club. Meets in Biggin Hill.

Lancs

North Lancs Computer Users Group
c/o M. Forham
14 Arundel Drive
Carleton Blackpool

...£1 pa. Includes Vics.

Vic Burnley
c/o John Ingham
72 Ardwick Street Burnley

...Proposed club.

Microcomputer Users
c/o Dave Walsh
19 St Margarets Way
Blackburn

...Includes Vics. £2.50 pa.

Leicestershire

Beaumont Leys Micro Club
c/o I. Lumb
103 Lime Grove Road
Beaumont Leys LE4 0UF

...£2.50 pa unwaged, £5 pa for others. Seems very active — three meetings a week! Mostly Vics.

Merseyside

Wirral Microcomputer Users Group
c/o Mike Keegan
205 Holmlands Drive Oxton
Birkenhead *£5 pa. Includes Vic*

Norfolk

Vic-20 Cromer
c/o J. Blair
7 Beach Road Cromer

...Proposed club based on local ICPUG branch.

Ireland

Irish Amateur Computer Club
c/o Martin Stapleton
48 Seacourt Dublin 3

...has a Vic section.

East Antrim Computer Club
c/o David Bolton
19 Carrickburn Road
Carrickfergus Co. Antrim

...£5 pa. David writes for us occasionally.

USA

Vic Users Group
c/o Paul V. Muffuletto
1206 N. Fraser Drive
Mesa, AZ 85203

California VIC Users Group
c/o Data Equipment Supply Corp. 8315 Firestone Blvd.
Downey, CA 90241

VIC 20 Software Exchange Club
2130 Colby Avenue
West Los Angeles, CA 90025

64 Users Group
c/o E. R. Kennedy
6029 Conta Costa Road
Oakland CA 94618

VICIMPET Users' Group
c/o Louis Roehrs
4 Waring Lane, Greenwood Village Littleton, CO 80121

VIC Users Club
c/o Edward Barszczewski
22 Tunxis Road
West Hartford, CT 06107

VIC Users Club
c/o Ray Thigpen
4071 Edgewater Drive
Orlando, FL 32804

Sun Coast VICs
c/o Mark Weddell
P.O. Box 1042
Indian Rocks Beach,
FL 33533

VIC 20/64 Users Support
Group
c/o David R. Tarvin
114 St. Clark Street
Pana, IL 62557

PET VIC Club (PVC)
c/o Paul Schmidt
40 S. Lincoln
Mundelein, IL 60060

VIC Chicago Club
c/o John L. Rosengarten
3822 N. Bell Avenue
Chicago, IL 60618

DuPage-64
c/o Gus Pagnotta
708 Prairie Avenue
Glen Ellyn, IL 60137

Eastern Massachusetts Vic
Users Group
c/o Frank Ordway
7 Flagg Road
Marlboro, MA 02173

VIC Users Group
c/o Ilene Hoffman-Sholar
193 Garden Street
Needham, MA 02192

VIC Users Club
c/o John Gannon
University of Michigan
School of Public Health
Ann Arbor, MI 48109

LIVE (Long Island VIC
Enthusiasts)
c/o Arnold Friedman
17 Picadilly road
Great Neck, NY 11023

VIC Users Club
c/o Michael Frantz
76 Radford Street
Staten Island, NY 10314

VIC Users Club
c/o Christopher
Kawasnicki
44 Harvey Avenue
Staten Island, NY 10314

Westchester County VIC
Users Group
c/o Joe Brown
P.O. Box 146
Pelham, NY 10552

VIC 20 User Club
c/o Jean F. Coppola
151-28 22nd Ave,
Whitestone, NY 11357

VIC 20 User Club
c/o Gary Overman
339 Park Avenue
Babylon, NY 11702

VIC Users Club
c/o David C. Fortenberry
Route 3, Box 351
Lincolnton, NC 28092

Nashville VIC Users Group
c/o Jane Maggard
American Computer Store
1004 8th Avenue, S.
Nashville, TN 37203

The VIClic
c/o Steve Graham
799 Ponderosa Drive
Sandy, UT 84070

VIC Users Group
c/o Dick Rossingnol
Rt. 2, Box 180
Lynchburg, VA 24501

VIC Users Group
c/o Donnie L. Thomspon
1502 Harvard Rd
Richmond, VA 23226

PET/64 Users
c/o Jerry Brinson
10136 E. 96th St
Indianapolis, IN 46256
317-842 6353

VIC Indy Club
c/o Ken Ralston
P.O. Box 11543
Indianapolis, IN 46201

VIC Connection
c/o Jim Kemp
1010 S. Elm
Henderson, KY 42420

VIC Interface Club
c/o C. Gary Hall
c/o Procter & Gamble Inst.
Shop, 780 Washington St.
Quincy, MA 02169

VIC Commodore Club
c/o Bill Kelley
2765 Bristol Rd
Dowling MI 49050

VIC-TIMES
c/o Thomas R. Molnar
46 Wayne St.
Edison, NJ 08817

VIC User Group
c/o Dr. Levitt
1250 Ocean Ave.
Brooklyn, NY 11230

VIC 20/Commodore 64 Users
Group
c/o Pete Lobol
31 Maple Dr.
Lindenhurst, NY 11757

VIC Users Club
c/o Tim Gromlovits
Rt. 11. Box 686
Hickory, NC 28601

64 User Group
c/o Keith Merrill
P.O. Box 28
Rhodes Road
Apalchin, NY 13732, USA



Vic/64 User Group
c/o Stephen McHoney
4912 Blarney Drive
Orlando, FL 32808, USA

Vic-20 User Group
c/o Wayne Lowery
5064 Bowdon Street
Marrero, LA 70072, USA

Vic 20 Users
c/o Dave DeCorso
324 N. 300 W.
Smithfield, UT 84335, USA

Northern Utah Vic & 64 Users
c/o David Sanders
P.O. Box 533
Garland, UT 84312, USA

Vic 20 Computer Club
c/o Michael D. Clark
947 N. Burroughs Ave.
Oak Harbour, WA 98277, USA

Commodore 64 Software
Exchange Group
c/o E. J. Rosenburg
P.O. Box 224
Oregon WI 53575, USA

TBH VIC-NIC's
PO Box 981
Salem
NH 03079, USA

National Vic-20 Users Club
PO Box 34575
Omaha
NE 68134, USA

...Newish commercial venture, basically a no-advertising quarterly newsletter at $10 pa.

Vic 20 Users Group
PO Box 459
Ladson
SC 29456, USA

...Looks like a newsletter-and-discounts operation with no meetings.

Vic-20 User's Group
c/o Roberto Morales, Jr.
655 Hernandez St.
Miramar
Puerto Rico 00907

Canada

Winnipeg Pet Users Group
Larry Neufeld
9-300 Enniskillen Avenue
Winnipeg, Manitoba
Canada R2V 0H9

...Reputedly includes Vic and 64 owners too.

Toronto Pet Users Group
c/o Chris Bennett
381 Lawrence Ave, W.Toronto,
Ontario Canada M5M 1B9

...Best of the Commodore user groups outside the UK, we hear (Butterfield is founder member). Lots of Vic activity, branches in Windsor, Flint and London (Ontario).

VIC-TIMS
c/o Greg Gross
2-830 Helena Street
Trail BC V1R 3X2 Canada

Australasia

Nelson Vic Users Group
c/o Peter Archer
PO Box 860
Nelson New Zealand

...Recent but keen. Trying to set up a worldwide network of Vic user groups to exchange public-domain software, club newsletters, etc.

Wonthaggi Computer Club
c/o Beth Tozer
52 Wentworth Road
Wonthaggi 3995 Victoria
Australia

Vic-Ups
c/o Eddie Mills
45 Zamia Place Greenwood 6024
Western Australia

...Big and enthusiastic (over 140 members, two meets per week).

Europe

Pet Benelux Exchange
Johan Scmilde Copytronics
Burg. van Suchtelenstraat 46
7413 XP Deventer Netherlands

...Highly organised, very professional. Lots of Vic/64 coverage.

Vic Club
c/o Guiseppe Jacob
Sardo Guiseppe via Ortensie
no. 8 57100 Livorno Italy

Vic-DK
Frederiks Allé 53
DK-8000 Aarhus C.
Denmark

...Keen and very active.

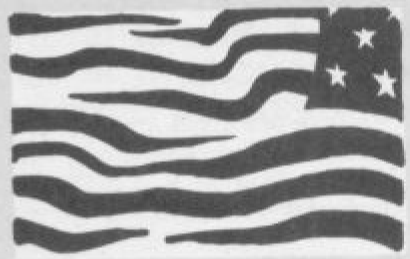
Microtel
87 Route du Polygone
67100 Strasbourg
France

...Includes a Vic section.

Vic Club
Nedre Bankegt 10
1750 Halden
Norway

Letter from America

by Mike Apsey



What happens in the States often presages what happens in Europe within the computer world. On the other hand, that is conspicuously *not* the case with some of Commodore's doings: on this side of the water, the Vic/64 business often seems to take a very different path ...

To check out what's going on over there, we've asked Mike Apsey, editor and publisher of *Journal/20,* to write us an occasional review of the US scene.

On this side of the pond the Vic is holding its own against the rivalry of a suntanned sibling, the Commodore 64; perhaps even Commodore is surprised. But other things are happening in America that shall be the topic of this letter.

Your intrepid author sports a bit of Vic bias; so 64 buffs reading this letter must bear with me, and keep in mind that an estimated 85 per cent of information specific to the Vic can be carried over to the machine you have.

Why do I prefer to play with the Vic? Well, I have a couple dozen responses, the selection of which is determined by the programming level of the reader or listener, and most of which are inappropriate to this column ...

64 software here is appearing in boxcar quantities. You can hardly pick up an American magazine without getting lost in the pushing and shoving of advertisements fighting against editorial space. There are magazines which could enter a new age of competition by dropping the text and admitting to being all ads. (*You sometimes get the same impression over here — Ed.*) One restraint is the US Post Office, which frowns on a publication obtaining special postal rates and then carrying more advertisements than text: provision exists for permit removal in such cases.

Sparkle

An American 64 problem at this time is in dealing with a rash of white dots which appear whenever the beast is processing. This screen rash is becoming known as "sparkle", and those who have sparkling 64s are getting impatient for a fix. Seems some units left Commodore with a missing heat sink, and addition of the sink is supposed to help.

Vic vs 64

Perhaps the real value of the 64 is yet to be realized. In the Vic arena, the 64 doesn't do much for me: in the case of the 64's ability to replace its ROM with RAM which lies dormant at the same memory addresses I am intrigued to say the least. Soon I expect to be seeing software that is booted into the 64 to change the operating system ... either slightly or mostly.

By loading another operating system, the 64 could disguise itself as anything. Even the restrictions of seven-bit ASCII could be eliminated, and eight-bit ASCII could be implemented.

More software

The American Vic is growing up. Several software houses have realized the market potential in developing serious multi-disk software for the Vic and are experiencing some success with more business oriented stuff.

Your **Wordcraft-20** has caught-on here, and is popping up in more households: **Practi-Calc** is storming American Vics as the greatest spreadsheet since VisiCalc: Mike Riley's **Flex-File** has its foot firmly on to the diving board with features that most Vic owners have never dreamed existed in any database, much less one for the Vic-20.

And by the time you read this, TOTL Software Inc. will have released a multi-disk business accounting package called **TOTL. Business 3.0:** I have been using the preliminary releases in managing my business affairs, and am suitably impressed with the collection of programs.

More disks

Many of the aforementioned goodies require (or support) the use of more than one disk drive, and the term 'twin-singles' is in use over here to describe the pair of 1540s or 1541s. Unfortunately, with the widespread attractiveness of adding a second drive over purchase of the traditional 'dual-drive', a semi-serious set of bugs is beginning to crawl out of the 1541 DOS.

It seems the bulk of twin-disk set-ups will hang unexpectedly and for no logical reason. There is no apparent pattern to the problem, and Commodore here is rumoured to be cold and unsympathetic regarding it.

Admittedly, getting through the ranks of secretaries at the main Commodore plant would be a good topic for a Scott Adams adventure: he could call it 'Find the President'. The object of the game would be to map out the maze of secretaries, departments, and misguided phone calls in search of the answer to a question.

At the end of the game, the president's office is found – but he isn't in, or was just fired. Good, huh?

Deals and dealers

Most of America's independent computer dealers are leaving the Vic-20 and 64 in highly publicized discontent. Many are even leaving Commodore's pair for machines they define as more profitable stock. Unhappy with Commodore's mass-merchandising moves into department stores and discount chains, the dealers, most of whom cannot purchase the Vic or 64 for what the chain stores sell them for, are quick to discourage onlookers from taking the two machines too seriously.

Those with high-volume purchasing power usually have low overheads and as a result remind me of a cattle auction barn. If you find what you are browsing for, there will be no-one to assist you. If you wish just to chat about programming, you quickly discover no-one knows how.

This is no way to market an item as complex as a computer, but people cannot resist the price and trust to luck to find a source of information about the beast.

Competition?

A recently retired high-ranking official of Texas Instruments, Computer Division was on television this week shown purchasing a Commodore 64. He prefered the 64 to the TI because of the combined features and peripheral affordability ...

22 vs 40 vs 80

There has also developed a group of Vic users experiencing good success with 40/80 column expansion devices. The discovery is centered upon the fact that when the screen card is in use, the space usually occupied by the moving Vic screen can be given back to Basic and the Vic neatly displays an even greater RAM capability (31,743 BYTES FREE). With nearly 32K available when used in this manner, the 64 only beats it by 6K.

Bye, bye, Vic?

There is a rumour here at the distributor and dealer level which places the 64 at $120 (*around £180, not so far away from the new UK price — Ed*) and kills the Vic at about Christmas time. It is unsubstantiated and has not affected sales of the little Vic; if anything, such rumours may just be wishful thinking on the part of those who have less to sell for more money.

Even if the Vic did stop falling off the ends of the production lines here at year's end, there are those who won't bat an eye. A couple of million have already been sold. Vics in the USA are sort of a computer version of the McDonald hamburger – nothing fancy, but extremely functional, and most surely a good value for the dollar ...

Regards from Stateside

An Epson on The 64?

ATTACHING A DECENT PRINTER

by Chris Durham

Once you've got a computer, a printer is probably the next major item that most people would try to acquire. Why? Well, you want hard copy (paper) for program development. And printers expand the horizons of a computer beyond playing games and the limitations of a small TV screen – as word processor programs come down in price, so more people think of using their computer to write letters (and articles!).

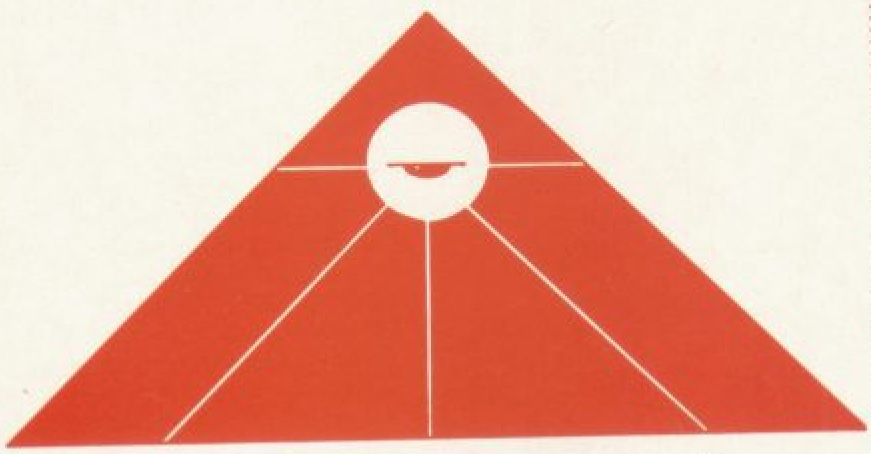
Epson has been producing dot-matrix printers for many years now to meet this demand and is definitely the market leader in printers for micros. Chris Durham takes a look at the latest offering from Epson – and finds out how easy, or difficult, it is to use with the CBM 64.

There are a large number of printers on the market, but only a few are specifically sold for use with Commodore computers.

This is partly because of the unique Commodore character set, with all those control characters and graphics; and then there's the fact that the main output port for a printer is based on either a parallel or serial form of the IEEE-488 standard.

Most non-Commodore printers either have a Centronics parallel or an RS232 serial connection, because that's what you *get* on most non-Commodore computers. In both cases an interface, or converter, is needed before you can use the printer with a Commodure computer.

Even those printers which do have an optional IEEE interface will not work directly with the Commodore 64, since the 64 (like the Vic) uses a funny serial version of the IEEE specification for the plug and socket rather than the full standard.



Why bother?

You may well be asking at this stage why anyone would want to go to all the bother of connecting non-Commodore printers at all. Commodore does after all provide a range of printers which plug in directly.

The fact is that the printers currently available for the Vic 20 and the 64 are very limited in their facilities: and if you want full word processing capabilities they all suffer from major drawbacks – like none of the cheaper Commodore printers have lower-case descenders (letters like 'j' and 'y' don't drop their tails below the line) and none of them have friction feed (which allows single pages to be fed through the printer).

If you really want high quality printing at a reasonable price you must look elsewhere. This is where Epson comes in.

The new FX-80 is by no means the only printer that can be interfaced to the Vic 20 or the 64. It is however the latest in a long line of Epson products that have proved themselves over the years. So, how well does the FX-80 live up to the reputation earned by its predecessors?

The printer's facilities

The FX-89 uses a 9×9 matrix of possible dot positions giving true descenders; it costs around £450 including VAT. It comes with sprocket pin-feed (for continuous stationery) and friction feed (for single sheets) as standard; but a fully adjustable tractor unit is extra. It takes standard widths of sprocket-hole paper (9.5 or 10ins), or 8.5ins plain paper (which can be on a roll as well as in single sheets).

The printer itself has a high-quality cream-coloured plastic case measuring about 16.5 by 13.5ins; it stands some 4ins high.

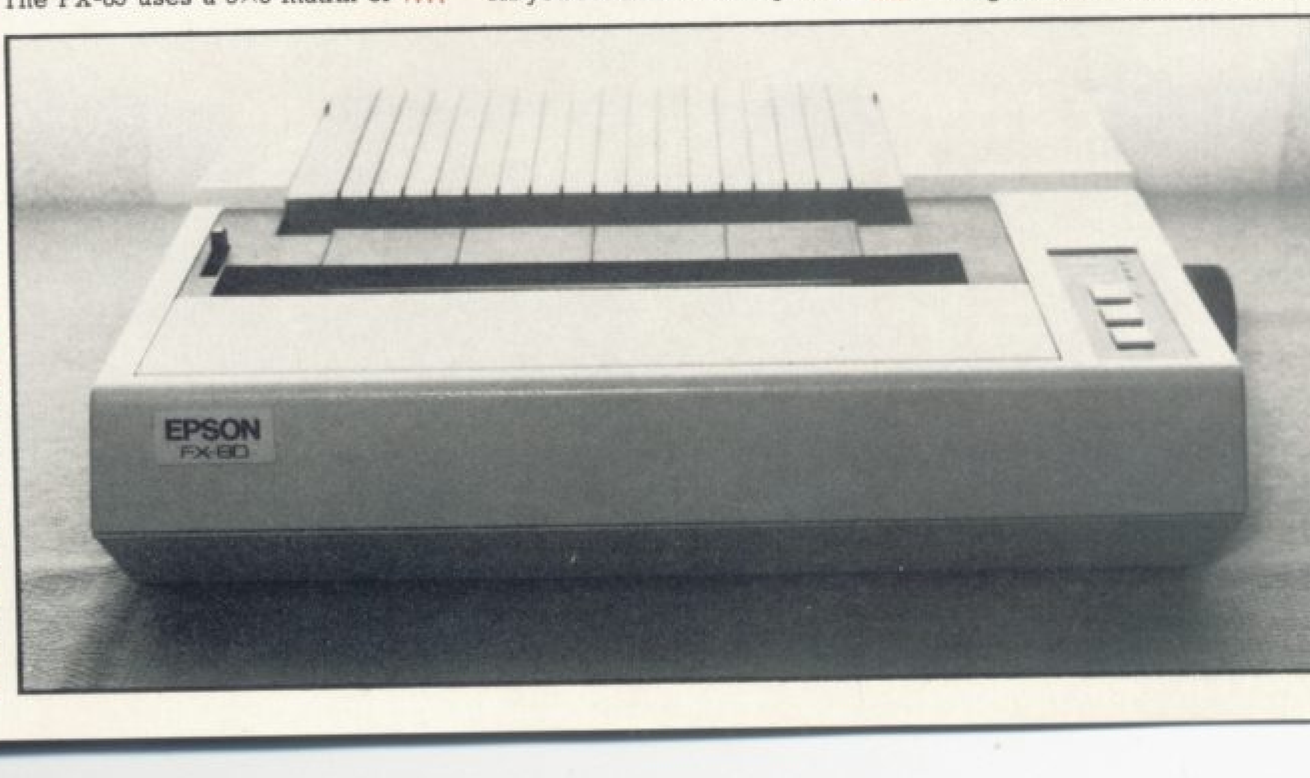
As you can see from the photographs there are only three main controls and four warning lights. The top control allows you to take the printer 'off-line' and then the other two controls give form-feed (moves a whole page though) and line-feed (moves up one line) respectively.

The warning lights show 'power on', 'ready'. 'paper out' and whether the printer is on or off line.

Extras

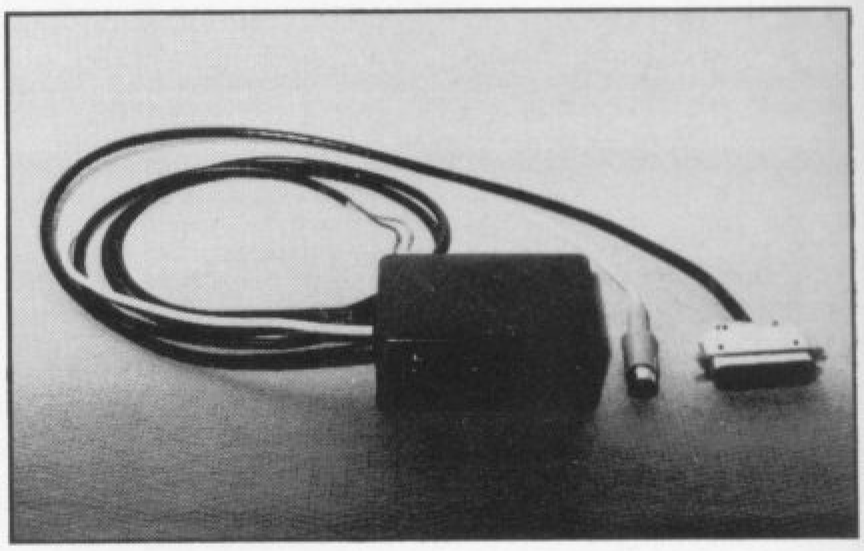
In addition to these main controls, however, there are a number of other facilities which may be selected by removing a small panel at the right hand side to gain access to two banks of little DIP switches. These facilities are shown in Table 1. Many of them are also program-selectable, so you can set some permanently and select others from your program for the particular printing task.

The switches are easy enough to get at, while being well out of the way of accidental resetting – a single screw holds the switch



Switch	Function	ON	OFF
1-1	Column length	132 chars	80 chars
1-2	Zero font	Slashed zero	Normal zero
1-3	Paper and detector	Disabled	Enabled
1-4	Input buffer	Enabled	Disabled
1-5	Initial print mode	Emphasized	Normal
1-6	International		
1-7	character		
1-8	set selection		
2-1	Printer Select mode	Fixed	Not fixed
2-2	Buzzer (bell)	Enabled	Disabled
2-3	Skip-over perforations	Enabled	Disabled
2-4	Auto line feed	LF and CR	LF from host

Table 1: DIP switch facilities



cover in place which is, quickly and easily lifted off without even having to move the paper. This has been well thought out by Epson, especially as some of its previous models required the removal of the entire outer casing to gain access to similar switches.

The printer is rated at 160 characters per second with a special 'quiet mode' printing at half that speed. Typically enough, like many of these manufacturer-supplied figures that print rate is not attainable in practice. Printing full 80 column lines with no spaces gives an average speed of only 100 cps with a single line being printed at 115 cps.

When the line width is reduced to 40 columns the average print speed drops to 75 cps – even with the bi-directional printing, which means the printer doesn't have to take the print head all the way back to the left-hand edge to start the next line.

All this means that in practical use for letters, listings and the like you'll average about 95 cps for an 80 column letter and 70 to 80 cps for a listing (where lines tend to be of differing length).

Although considerably less than the rated speed, this is still very fast – especially when you realise that a printer like the Vic 1515 averages a mere 15-20 cps.

One very big plus for the printer is the standard 2K buffer which is available to your output if you are not specifying your own character set. The buffer is there because the computer can shove stuff to be printed to the printer much faster than the printer can print it.

Given this 2K chunk of memory, the printer can take virtually the whole page of a shortish letter all at once – leaving the computer to get on with further work while the printer goes ahead with the job of printing whatever's in the buffer.

At the speeds quoted for practical use this means you save approximately 20 seconds per document. It may not seem much, but those 20 seconds soon mount up.

Versatility

It is when you consider the vast range of printing styles and printing options available to you that the real versatility of the FX-80 begins to show.

There are two main type fonts, Pica at 10 characters per inch and Elite at 12 per inch – they are more or less similar to the standard type styles on portable and other typewriters. Both of the type styles can be printed in italics, both can be enlarged and both have corresponding sub- and superscript modes.

Add to that the fact that characters can be 'double-struck' for bold printing and in some cases 'emphasised' as well: and you begin to see that the permutations for pretty printing are considerable.

For graphics there's a wide range of commands to allow varying densities of plotting giving better resolution than most TV screens! There are comprehensive horizontal and vertical tab commands and the paper can be moved both up and down under program control. As if that were not enough to be going on with, you can download your own character set with the ability to switch back and forth between that and the normal one – together with a number of international character sets which may be selected as well.

All in all there are a total of 73 control commands which can be sent to the printer to gain access to all these functions. But they aren't hard to get at: figure 1 shows how easy it is to use these codes, and also gives a selection of the printing options.

There is not room to list every function, but others not previously mentioned include setting of right and left margins, length of form, backspace, and sounding the bell . . . they're all in the manual supplied with the printer, a booklet that is well laid out showing each individual command with examples. It has a ready-reference section at the back which ensures that however infrequently you use some of the facilities you will have little difficulty in finding the relevant section in the manual. (Computer manufacturers could do much worse than follow this example of a useable manual . . .)

Connecting the printer

The FX-80 has a Centronics parallel connection as standard. So for the 64 it is necessary to use some type of interface.

There are two main types



available: one connects the printer from the user port with software conversion, the other goes from the normal serial port with a hardware conversion.

The first of these involves loading a small program before you can load or run any other program: this sets up the user port as a parallel printer port, and the printer is then connected by either a ribbon cable or a multi-core cable directly to the port. The software also carries out any conversion necessary to allow the normal upper and lower case ASCII character set to be printed. (The Commodore character set is slightly different to allow for the graphics on a Commodore printer.)

The hardware interfaces work by accepting the data on the normal serial cable and converting it to the Centronics parallel format before sending it on to the printer. They too do the necessary character conversion, but they can produce some additional conversions as well – changing the control characters into meaningful alphanumeric strings, for instance.

Interfaces in use

I looked at a sample of both types of interface to see how they performed in practice.

The first point to bear in mind if you're choosing a software interface is that the program will occupy an area in memory.

That won't matter too much from the overall point of view of space since the 64 has a large memory to start with. But there may be a problem if you want to use your printer in conjunction with other programs or cartridges: these may well try to occupy the same area of memory as your printer program. The Simon's Basic cartridge, for example, will not work with any software-based printer interfaces which use the RAM area from 49152-53247 – or more accurately, the cartridge *will* work: but the printer won't! Not all the programs do use this area, so it's worth checking before you buy if you intend using the printer in conjunction with ROM-based cartridges.

Once the converter program has been run you access the printer in exactly the same way as a normal Commodore printer.

The two software interfaces I looked at were from Vizasoft and Stack. The **Vizasoft** interface is not generally available since the company sells it with a cable primarily for users of its word processor Vizawrite – the Centronics interface software is included free on the word processor disk. Visasoft might be prepared to supply both cable and software for about £26 if you can convince them you are desperate.

The **Stack** interface is a multi-core cable with the software coming on a cassette. There were two problems with the Stack interface, the first of which is that the cable is very inflexible and the plug that goes into the User-port is prevented from seating correctly by a badly-placed screw head. The result is that if you move your computer the plug promptly comes out of the user port. A ribbon cable would have been far superior even if not quite so robust.

The second drawback to the Stack solution lies in the software. When you initialise the program it resets the main memory pointers – thus losing any Basic program in the computer. You might try to recover from a program that has crashed by using RUN/STOP and RESTORE; but that will also turn off the printer program and there is then no way of using the printer again without losing the Basic program. The Vizasoft program had no effect on any Basic program when re-initialised: so it *can* be done.

The Stack cable and program costs £33.35, but I can't honestly recommend it in its current form.

A ribbon cable and cassette-based program are also available from **Audiogenic;** this costs £27.90 and would appear to be similar to the Vizasoft version, but was not actually tested.

Hardware interface

The two hardware interfaces I looked at were from **Ibek Systems** and **RAM Electronics.** The **Ibek** unit can be seen in the photograph; it costs £71.24 and consists of a black plastic box with three leads. One plugs into the mains, the second plugs into the serial port on the computer or if you have disks, into the back of the disk drive; the third goes to the printer. There is a small switch on the box to allow character conversion if required.

Inside there are three other straps which can be resoldered to change the device number of the printer from 4 to 5 and to allow automatic sending of carriage return/line feed if the printer requires it.

All in all it seems odd that one strap has been taken to a switch and others haven't. The manufacturers say that the other functions alter only occasionally and therefore switches are unnecessary; I disagree, since the CR/LF option would be useful if it were more accessible. But it would not be a long job to add the extra switches yourself if you need them.

The interface worked well when printing, but the initial version I tried caused the disk drives to hang up repeatedly when accessing files. A second unit worked without any problem, though, so this would appear to have been an isolated fault.

The second hardware interface was the Vicsprint from **RAM Electronics;** this costs £57.50 and consists of a cartridge which plugs into the expansion slot with one cable attached to the serial port and a ribbon cable connecting to the printer.

The only reason the unit needs to plug into the expansion socket is for power, for all the data comes via the serial port connection. The serial lead is long enough to plug into the back of the disk drive if required, but the ribbon cable is under a metre and that is really too short; at least another 12 inches is needed to enable the cable to be routed tidily instead of having to take the shortest route.

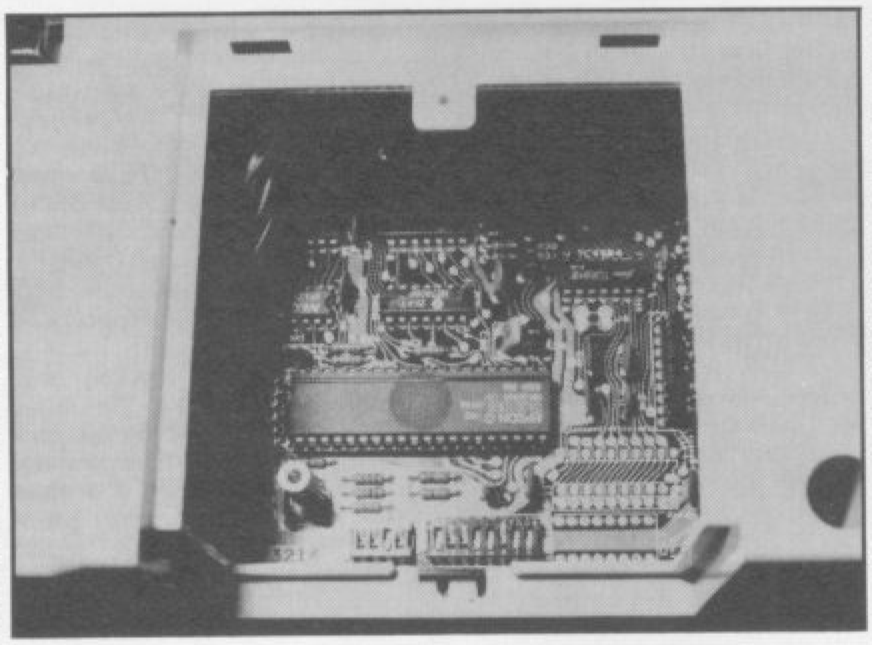
The interface carries out all necessary conversions plus adding CR and LF if needed. The secondary address used when opening the printer channel determines which options are selected. One excellent option is the facility to convert the Commodore control characters into meaningful strings; (RED) is much more understandable than a reversed pound sign!

There is one major drawback with this system; those of you

who have tried to use an unmodified 1540 disk drive with the 64 (rather than the newer 1541) will know that there is a timing problem which causes the system to hang up. Unfortunately the Vicsprint also suffers from the same problem. It only happens if you try to use 'PRINT#'; after a while the system just stops working. Listings and 'PRINT' after 'CMD4' are unaffected.

To be fair, the suppliers do point this out on the instruction sheet and explain how to avoid the problem; the screen has to be turned off when you access the serial bus. This works, but it is not exactly an elegant solution – and it will not be possible if you have a commercial program that cannot be altered. Hopefully Commodore will release the necessary information to enable RAM to overcome this problem (note that the Ibek interface does *not* suffer from the same fault).

There are versions of Vicsprint for the Vic-20 and the 64, with a new implementation



appearing shortly which plugs into the Epson FX-80 directly and draws its power from there. This means that the only connection is a single cable to the serial port – a very neat solution. I hope to try it out when it's available and let you know.

```
10 REM ********************************************
20 REM PROGRAM TO DOWNLOAD COMMODORE CHARACTER SET
30 REM TO AN EPSON FX-80 PRINTER - BY CHRIS DURHAM
40 REM ********************************************
45 POKE 52,152:POKE 56,152:CLR:REM RESERVE SPACE IN MEMORY FOR CHAR SET
47 POKE 53280,14:POKE 53281,6
50 PRINT"(CLR HOME)(CUR DN)(CUR DN)  SWOP CHAR SET INTO MAIN MEMORY"
60 CS=53248:CL=CS+512:ML=38913
70 PRINTCHR$(142):REM SWITCH TO UPPER CASE
80 POKE 56334,PEEK(56334) AND 254:REM TURN OFF KEYSCAN INTERRUPT TIMER
90 POKE 1,PEEK(1) AND 251:REM SWITCH IN CHAR SET
95 FOR A=0 TO 511:POKE ML+A,PEEK(CL+A):NEXT A:REM TRANSFER CHARS
100 ML=ML+512:FOR CH=1 TO 27
105 READ X:FOR A=0 TO 7
110 IF CH<25 THEN POKE ML+A,255-PEEK(CS+(X*8)+A):REM TURN INTO REVERSED CHARS
115 IF CH>=25 THEN POKE ML+A,PEEK(CS+(X*8)+A): REM CHARS NOT IN EPSON SET
120 NEXT A:ML=ML+8:NEXT CH
125 POKE 1,PEEK(1) OR 4:REM SWITCH IN I/O
130 POKE 56334,PEEK(56334) OR 1:REM TURN ON KEYSCAN INTERRUPT TIMER
135 PRINT"  CONVERT CHARS TO PRINTER FORMAT(CUR DN)"
137 DIM B1(8):FOR A=0 TO 7:B1(A+1)=2^A:NEXT A
140 PL=39700:MP=38913
145 FOR Y=PL TO PL+546:POKE Y,0:NEXT Y
150 FOR Y=PL TO PL+540 STEP 6
160 FOR A=7 TO 2 STEP -1
170 FOR B=0 TO 7
180 IF (PEEK(MP+B) AND B1(A)) THEN POKE(Y+7-A),PEEK(Y+7-A) OR B1(8-B)
190 NEXT B,A:MP=MP+8:NEXT Y
200 OPEN4,4
210 REM TRANSFER EXISTING EPSON CHAR SET TO USER AREA
215 PRINT#4,CHR$(27);"R";CHR$(0);:REM SELECT USA SET
220 PRINT#4,CHR$(27);":";CHR$(0);CHR$(0);CHR$(0);
225 PRINT"  NOW TRANSFER COMMODORE CHARS(CUR DN)"
227 FOR L=1 TO 2:READFC,LC
230 PRINT#4,CHR$(27);"&";CHR$(0);CHR$(FC);CHR$(LC);
235 FOR CH=0 TO 31:PRINT#4,CHR$(139);
240 FOR A=0 TO 4
250 PRINT#4,CHR$(PEEK(PL+(CH*6)+A));:PRINT#4,CHR$(0);
255 NEXT A:PRINT#4,CHR$(PEEK(PL+(CH*6)+5));
260 NEXT CH:PL=PL+(32*6):NEXT L
262 REM ALLOW ALL ASCII CODES (0-255) TO BE PRINTABLE
264 PRINT#4,CHR$(27);"I";CHR$(1);CHR$(27);"6";
266 PRINT"  NOW TRANSFER CONTROL/COLOUR CODES(CUR DN)"
268 REM ALSO INCLUDES CHARS NOT IN STANDARD EPSON SET
270 PL=40084:FOR CH=0 TO 26
280 READ CP
290 PRINT#4,CHR$(27);"&";CHR$(0);CHR$(CP);CHR$(CP);
300 PRINT#4,CHR$(139);
310 FOR A=0 TO 4
320 PRINT#4,CHR$(PEEK(PL+(CH*6)+A));:PRINT#4,CHR$(0);
325 NEXT A:PRINT#4,CHR$(PEEK(PL+(CH*6)+5));
330 NEXT CH
335 REM SWITCH TO USER DEFINED CHAR SET
340 PRINT#4,CHR$(27);"%";CHR$(1);CHR$(0);
350 PRINT#4,CHR$(27);"E";:REM SET EMPHASISED MODE
360 PRINT#4:CLOSE4
370 PRINT"(CLR HOME)(CUR DN)(CUR DN)(CUR DN)       CHAR SET TRANSFER COMPLETE":PR
INT
375 PRINT"       COMMODORE CHAR SET SELECTED(CUR DN)"
377 PRINT"****************************************(CUR DN)"
380 PRINT"TO SELECT EPSON CHAR SET, TYPE:"
390 PRINT"(CUR DN)  (YEL)PRINT#4,CHR$(27);"CHR$(34)"%"CHR$(34)";CHR$(0);CHR$(0);(
WHT)(CUR DN)(CUR DN)"
400 PRINT"TO RE-SELECT COMMODORE CHAR SET, TYPE:"
410 PRINT"(CUR DN)  (YEL)PRINT#4,CHR$(27);"CHR$(34)"%"CHR$(34)";CHR$(1);CHR$(0);(
WHT)(CUR DN)(CUR DN)"
420 PRINT"ENSURING STREAM 4 IS OPEN FOR PRINTER   OUTPUT.
430 END
1000 DATA80,5,28,95,92,30,31,94,65,85,86,87
1010 DATA88,89,90,91,18,70,83,19,81,17,66,29,28,31,94
1015 DATA192,223,160,191
1020 DATA144,5,28,159,156,30,31,158,129,149,150,151
1030 DATA152,153,154,155,18,146,147,19,145,17,157,29,92,95,255
```

Downloading the Commodore character set

The main advantage of a Commodore printer is that all the graphics and control characters are available for listings and printouts. Most other printers either print nothing or something funny like Greek letters; neither makes reading the resulting printout very easy.

By using the Epson's ability to download a user-defined character set, however, we can make the FX-80 print *all* the Commodore characters.

What is needed is a nifty bit of programming to pass the ROM-based characters from the computer down to the printer. This is not quite as easy as it first appears – the computer builds up the characters row by row; the printer builds them up column by column. If you try passing the data for the character set as it is held normally, you would end up with all the characters lying flat on their backs!

Program 1 shows how to convert the characters to appear the right way round and how to download them to the Epson printer. The program has a number of remarks put in as screen messages; these double as progress reports since the program takes a couple of minutes to run. You only need to run this once at the start of a session, unless of course you turn the printer off by mistake.

Instructions are included on how to select either the standard Epson character set or the Commodore set; this can be done either from within a program or directly from the keyboard.

There are a number of points to note before using this program. First, the maximum number of adjacent horizontal dots in a printer character is six. Some Commodore characters, like the heart and the spade, use seven horizontal dots on the TV screen; these will be truncated when printing. The only way to avoid this is to design your own characters for these symbols.

Second, there seems to be no way of replacing the printer control codes in character positions 18-20 inclusive. This means that the character code for HOME, REV ON and INSERT cannot be printed since they occupy these character positions in the Commodore ASCII set. Next, because both upper and lower case characters are held in the printer, there is not room for reversed characters as well. Finally, the 'zero font' switch on the printer must be set to the off position.

Within these constraints this program should at least provide readable listings.

In conclusion

The Epson FX-80 is a versatile, efficient printer capable of producing decent-quality printing at a reasonable price. The additional facilities it offers over the cheaper Commodore printers make it well worth the effort of finding a suitable interface.

Even if you cannot afford the FX-80, its cheaper stable-mate the RX-80 may still bear looking at – although you will not be able to obtain the Commodore characters.

There are a number of interfaces on the market at present to attach Commodore computers to Epson (and other) printers – but you should be careful when using a software based interface if you wish to use your printer in conjunction with add-on cartridges, because of possible memory clash problems. The hardware interfaces, although more expensive, seem a better general solution; the interface that will actually fit inside the printer should be the most practical of all when it appears, provided the timing problem has been overcome.

Don't forget that all the interfaces mentioned are general Centronics converters and should work with Centronics parallel printers. Do make sure you check first with your supplier though, to ensure compatibility.

Commodore 64 Computing

by Ian Sinclair

Commodore user manuals leave plenty of scope for independent writers to fill in the details, and Mr. Sinclair has been quick to cover the 64. The book is a paperback of some 130 pages and the preface promises a serious approach to BASIC programming with a bias towards data processing for business applications, but is this promise met?

The first chapter starts with detailed instructions on how to set up the 64 and tune in the television. This is followed by the usual tour of the keyboard (yawn) and details of loading and saving to tape and disk.

There's not much in this chapter that isn't in the first 20 pages of the User Manual, except a comment concerning use of the RUN/STOP and RESTORE keys. These keys must be "tapped sharply" to restore the program. Until I read this I thought the contacts on my RESTORE key needed cleaning!

The second chapter starts the reader's instruction in Basic beginning with immediate mode and the ubiquitous PRINT statement. This section deviates from the User Manual by also introducing the TAB and SPC commands and their use in positioning text and centering titles. The different types of variables are covered, but we picked up a flaw – Ian Sinclair says the 64 works on integer variables faster than floating point. This is incorrect; all Commodore computers convert integer variables into floating point before operating on them and change them back into integer format afterwards so there is no increase in speed. Integer variables only save memory, not processing time.

Good on DEF

Data input is then covered with the use of INPUT, READ-DATA and GET statements accompanied by good advice on preventing incorrect data entry by the use of 'mug traps.'

All good stuff, from where the book moves on to arithmetic operations and their priority. This discussion includes the oft-forgotten function definition (DEF FN), very useful for repetitive calculations like VAT or rounding-up operations and normally neglected by book writers. The remainder of the chapter covers string manipulations (LEN, VAL, LEFT$ etc) and uses several useful practical examples to display their effectiveness – GETA$ instead of GETA is encouraged, and we'll go along with that as a way to avoid incorrect data input.

Chapter four leads on to branching and loops. Rounding errors and their effects on conditional branching is explained: the use of FOR-TEXT loops for timing, counting and repeating operations leads on logically enough to subscripted variables, although the fact that subscripted *integer* variables take up much less space than subscripted *floating-point* variables is not emphasised. Matrices and data storage finishes this chapter using magazine articles and their contents as examples.

'Data processing' is the subject of Chapter Five, and again that flows on smoothly from the previous one. The use and design of menus in data processing programs is covered, and ON-GOING and ON-GOSUB statements are introduced. Together with that section on the GET statement, this reinforces the preferences, prejudices and lessons learned in previous chapters: hooray for structured programming!

'Total' program design using the top-down approach is discussed, and Ian Sinclair stresses the need for standard subroutines. The overall feeling is that disciplined programming is A Good Thing, and it is.

So far so good.

Total design

'Total' program design using the top-down approach is discussed, and Ian Sinclair stresses the need for standard subroutines. The overall feeling is that disciplined programming is A Good Thing, and it is.

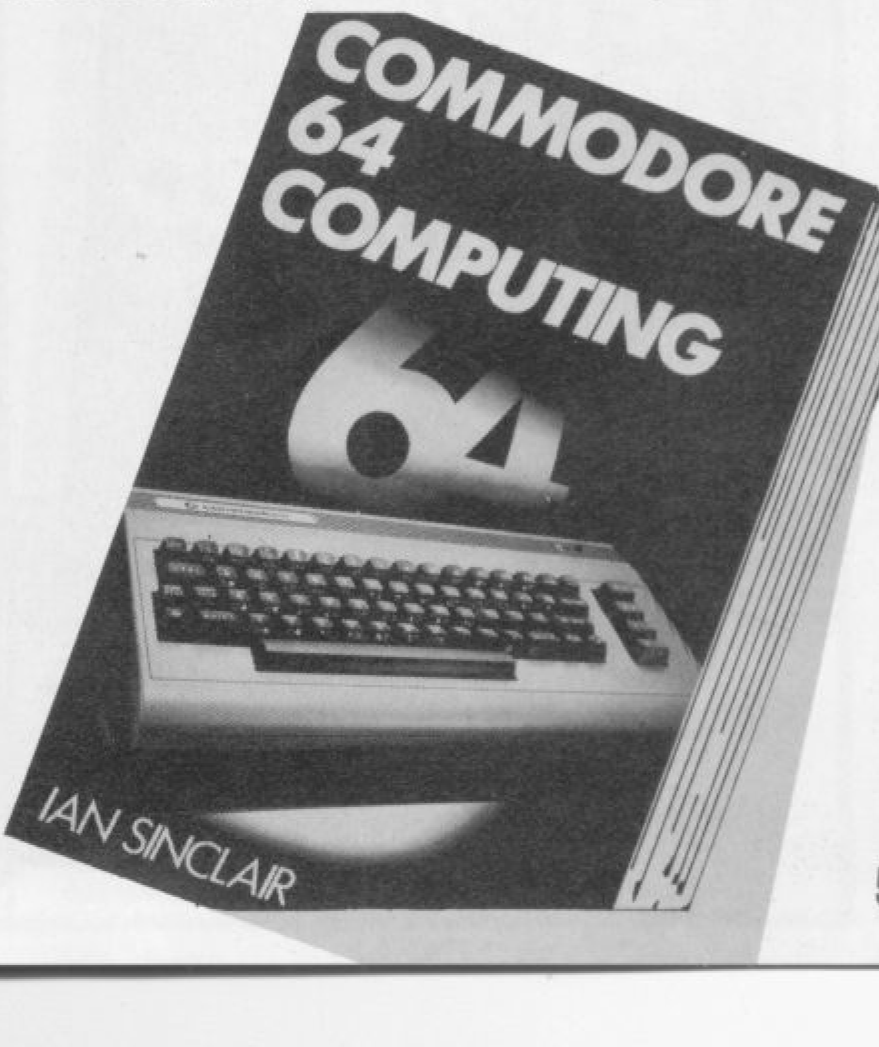
So far so good. The sixth chapter, 'Introducing Graphics', prompted thoughts of hi-res and multicolour plotting: but it was not to be. This chapter deals only with the graphics characters available from the keyboard and how they can be combined into strings to draw titles, borders and bar charts. How disappointing.

At least Mr Sinclair follows his own implied views about discipline. When reading a Commodore program listing it is often difficult to remember which key represents which graphics symbol without scanning the whole keyboard, and sometimes the number of spaces required can be misleading. Ian Sinclair overcomes this problem by using the *Computing Today* standards for special function keys: *Vic Computing* and *Commodore User* are also introducing their own standard to make program listings more legible.

Animation using strings is briefly covered here, and the POKE command is introduced together with some good advice about SAVEing a program with POKEs in it before RUNing it in case the computer crashes. Practical rather than earthshaking stuff.

Sprite disappointment

In fact the chapters up to now have covered *any* Commodore



computer, not just the 64. Chapter Seven, however, deals with sprites: and instantly there are problems. The sectic n on sprite priority is confusing — and in parts incorrect.

For instance, Sinclair says that POKEing 0 into address 53269 (the sprite enable address) will give the sprite level 0 priority: in fact it will disable all sprites.

Again, moving sprites is covered: but the origin of a Sprite is given as its bottom left corner when it is actually the *top* left.

Multi-colour sprites are mentioned: but Ian Sinclair attributes this only to sprites 0 and 1 — when all *eight* sprites can be multi-coloured. Nor does he explain how to use the multi-colour registers to create such sprites.

In addition to its graphics capabilities, the other starring quality of the 64 is the now-famous SID synthesiser chip. 'Function Keys and Sound Generator' comes next. Well, the section on function keys isn't worth much: and the piece on sound capabilities are limited to warning notes to prompt responses during DP programs — an application which hardly strains the best synthesiser chip currently implemented in a home computer.

At least there is a useful table which gives the real duration time against attack, decay and release settings.

The ninth chapter is a miscellaneous collection of the commands not yet covered, like FRE, SGN, ABS and SQRT. The main theme is on debugging programs by using the STOP and CONT commands and by printing variables or dumping them to tape to check that they are consistent with those expected. You also get the 64's real-time clock and its uses for alarms, timers and the like: and a useful-looking digital clock subroutine is included.

The final chapter deals briefly with peripherals, such as printers, joysticks, digitisers and light pens and their applications. The cartridges like Simon Basic, Pascal, and Forth (which are available or not as the case may be) are also mentioned just to whet your appetite.

The one and only appendix gives a rather incomplete set of Basic equivalents, comparing Commodore's statements to ZX and Atom computers so you can convert programs.

Lord knows why you should want to run a ZX81 program on a 64, still less a 64 program on a ZX81. A Vic-vs-64-vs-Pet converter might have been more useful, or perhaps a Spectrum-Dragon-BBC conversion chart.

Overall impressions? The book is technically good. It's well-written, readable and logically presented — you don't need to sit hunched over your computer, book in one hand while you stab the keyboard with the other. The example programs are short and professional, which reduces typing time and the attendent possibility of errors.

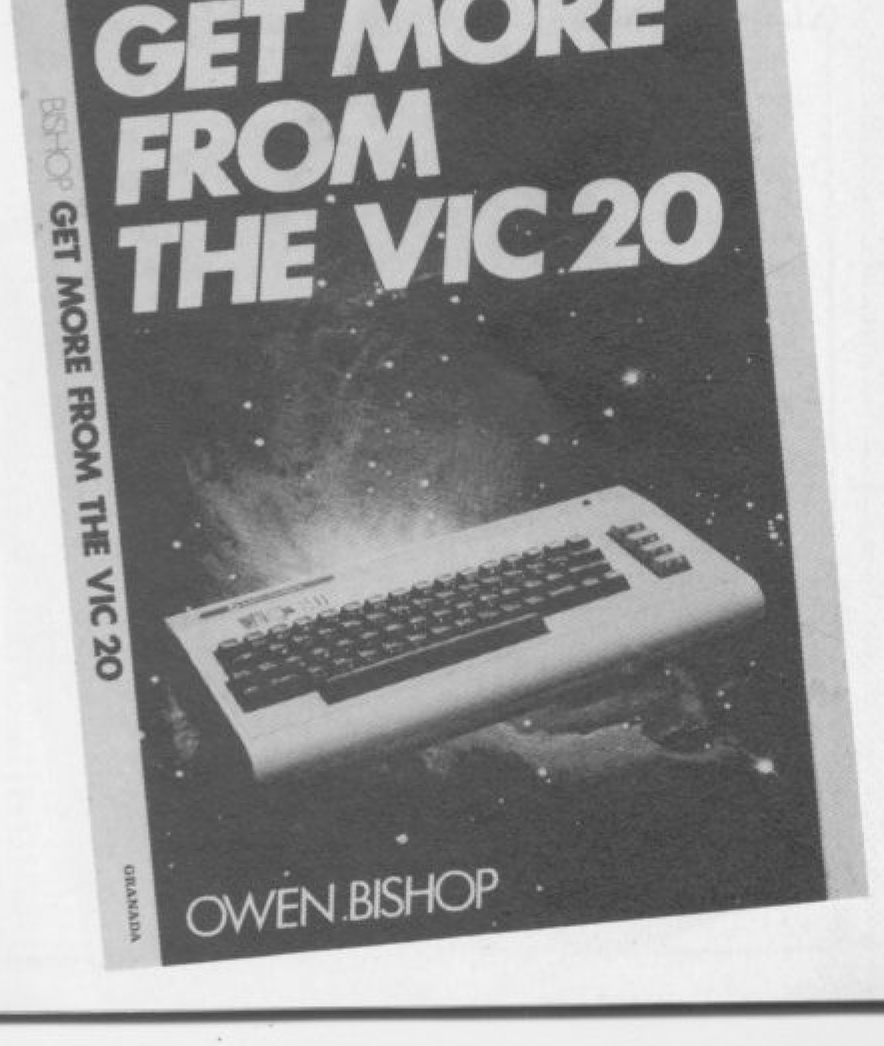
The promise of a data processing bias is met, but the book deals mainly with the input and output of data and not the important bit in the middle — ie the data processing. The nearest we get is a crude alphabetic sorting routine.

But I think the title is misleading. "Commodore 64 Computing" conjours up thoughts of high-resolution multi-colour graphics with sprites, user-definable characters, horizontal and vertical scrolling, better-than Moog music synthesis, and clever ways with joysticks. Hardly any of these features are covered in depth: the bulk of the book is general and could be applied to any Commodore computer.

If you're a game/graphics/sound freak, this book is not for you. If you are in business and want to have a crack at data processing, the book offers a sound introduction to the subject. But in general it sadly falls well short of being a complete work.

Commodore 64 Computing By Ian Sinclair
***Pı·blished by** Granada Price £5.95.*

Get More From the Vic?



Owen Bishop welcomes us to the audio-visual paradise of the Vic 20 and stresses above all that "computing is fun". I'm sure he is not suggesting that home computing *wasn't* fun before sound and colour, although one gets the impression that the book's attitude is one of coaxing indolent teenagers away from video-game parlours and the television and on to the Vic keyboard ...

The first two chapters are rather simplistic – reassuring perhaps to a first time user who has not read the manual, but rather irritating to anyone else. We are told how to unpack the box and connect the plug.

The rest of the book could be divided into three parts; colour graphics, applied mathematics and sound. Owen Bishop chooses to separate these sections into nine chapters – largely, I suspect, to make it easier to read (although the chapter titles 'More About Graphics' and 'More About Sound' do nothing to enhance this illusion).

In between PEEKing and POKEing colour and sound out of RAM, we are introduced to the major functions of the keyboard and taught how to use variables, arrays and various programming techniques. The functions are well applied to numerous entertaining programs: there are useful programs too to design your own colour characters and invent sounds and animated graphics.

For the most part the listed programs are good, although yet again the graphics characters do not come out very well on the printed page. Good games programs are conspicuously absent ... and it may surprise nobody to learn that Mr Bishop is also the author of *The Vic 20 Games Book.*

The book ends with some useful hints on memory saving and writing faster programs; and the appendix contains some potentially helpful tables.

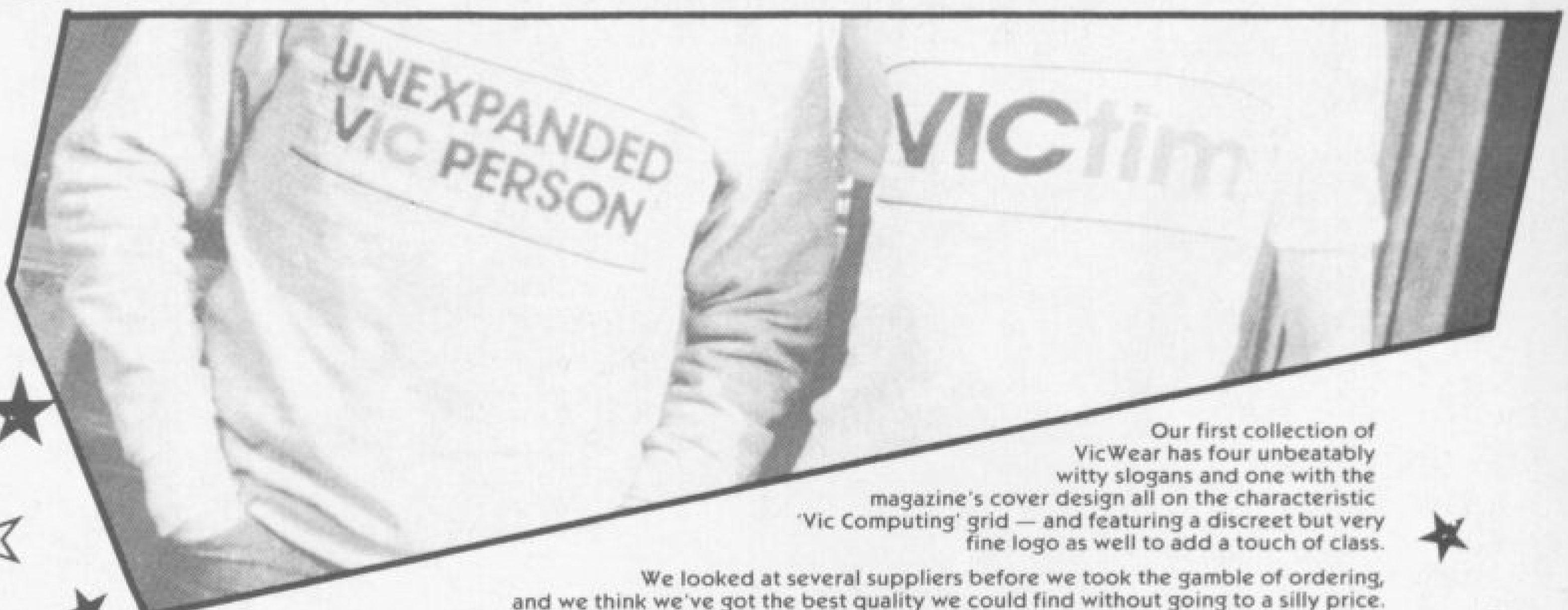
Generally speaking, the book is somewhat padded out to justify the £5.95 price tag. It has some useful material though, it is well structured and illustrated, and it is written in a fairly readable style.

Richard Schwarz

Get More From the Vic 20
By Owen Bishop
published by Granada
Price £5.95

Vic Wear

GOOD QUALITY TEE SHIRTS AND SWEATSHIRTS FROM VIC



Our first collection of VicWear has four unbeatably witty slogans and one with the magazine's cover design all on the characteristic 'Vic Computing' grid — and featuring a discreet but very fine logo as well to add a touch of class.

We looked at several suppliers before we took the gamble of ordering, and we think we've got the best quality we could find without going to a silly price. They are some kind of cotton mix, which we're told are less prone to shrink. They feel nicer, too.

- **Plain white tee-shirts** with printing in black and a bit of red: very smart, very comfortable for playing games, writing programs, and/or reading 'Vic Computing'. Price £3.00
- **Blue tee-shirts** with printing in black and red: we have to pay extra for the colour, but they do look nice. Price only £3.20
- **Grey tee-shirts** with printing in black and red: a bit thicker and again we have to pay more, so you do too. But still good value. Price £3.40
- **Grey sweatshirts** with printing in black and red: looks great. Excellent quality, warm but not too heavy. Price £7.00

Sizes: **small**(32-34), **medium** (34-36), **large** (38-40), and **extra large** (42-44 or more). The sweatshirts are pretty baggy — they actually make rather fine dresses on ladies of more diminutive stature.

Kids: We didn't get much demand for children's sizes the first time around. But 'small' seems to fit 8-9 year olds.

Overseas prices: Those prices above are for UK only. for **other European countries** add 30p per tee-shirt and 70p per sweatshirt. **For everywhere else in the world** add 95p per tee-shirt and £2.92 per sweatshirt. (It's the extra postage charge, of course.)

Extra charges: There aren't any, it's all included — VAT, post, packing, etc.

How to order: If you have an Access card, you can order by phone — call **01-241 2448.**

Otherwise fill in the form. Print a name and address on the despatch label (we cut that off and use it on the parcel), tot up the amount, and enclose a cheque or postal order with the order or give your Access number (for that we do need your signature — and make sure the address given is the one that Access has for you). Return the form to **VICWEAR, The Metropolitan, Enfield Road, London N1 5AZ**

How long to wait: Orders are processed as they arrive. Allow 28 days for delivery, then phone us to complain.

ORDER FORM

Write the number you require in the relevant Boxes	T-SHIRTS White S	M	L	XL	Blue S	M	L	XL	Grey S	M	L	XL	SWEAT-SHIRTS Grey S	M	L	XL
A. VIC COMPUTING																
B. CAPTAIN KILOBYTE																
C. VICTIM																
D. POKE 36879																
E. UNEXPANDED VIC																

Please state second choice in case of unavailability

White T-shirts £3.00 Grey T-Shirts £3.40
Blue T-Shirts £3.20 Sweat-Shirts £7.00
Uk prices only. See previous page for overseas postage charges.

TOTAL £______
Please make cheques or postal orders payable to 'Vic Computing'

Your Name

ADDRESS

Access Card No:

Date Signature

THIS IS YOUR LABEL — PLEASE PRINT CAREFULLY

TO

TOMMY'S TIPS



Dear Tommy, **Reading the June 1983 issue, I saw that someone asked if it is normal for the TV to buzz and change its tone as the computer is working. I also found this buzzing happened when I was typing in MOTHER*HEN from the February issue: as I was typing, the buzz got louder. When I had finished typing the program I cleared the screen and then I realised that the buzz had stopped. Then I typed LIST and pressed RETURN. The program came through on the screen, but the buzzing started up again.**

The buzzing is caused by part of the video signal getting into the sound channel of the TV. This can be caused by three things:

- Your TV set needs aligning. This is an internal operation and can only be carried out by a skilled technician with a certain amount of gear.
- Your Vic is not set up properly. Up to a point the same comment applies and so the safest option is to contact your Vic dealer.
- The most likely cause is that the TV needs to be tuned to the Vic more accurately. The details of how to do this vary from TV to TV, but is usually very easy.

If you get the same result on a different TV, this would tend to point to a problem in the Vic; but on the other hand if both your TVs are a bit long in the tooth then they might *both* need realigning.

Dear Tommy, **Please could you tell me how I can change the Vic's cursor.**

It is not easy to alter the Vic's cursor successfully. Some computers use a cursor generated in hardware, and you can select a flashing block (like the Vic has as standard): or an underline: or whatever you fancy. On the Vic, however, the cursor is generated in software.

What happens is that 60 times a second the processor receives an interrupt signal from the hardware which forces it to execute a part of the ROM called the 'interrupt service routine' (this starts at $EABF). The ISR updates the jiffy clock, blinks the cursor if there is one being displayed, and scans the keyboard.

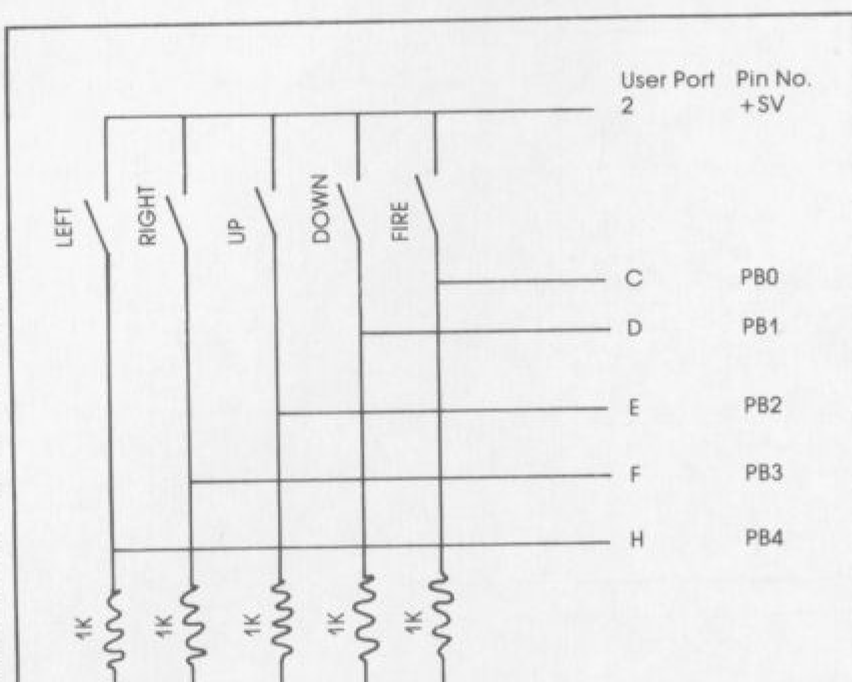
The 'cursor' is in fact produced by switching the character under the cursor between normal and reverse video. The only real change you could make is to make the cursor a constant reverse character, or maybe a character flashing on and off. The only way to do this is to write a routine in assembly language, update the jiffy clock (JSR $FFEA) and then jump to $EAEF. You would then have to alter the interrupt vector in $314 to point to your routine.

Dear Tommy, **I am trying to utilise an old calculator keyboard as a joystick-type control for a disabled friend of mine. Could you please show me how to wire in a straight five-switch device with left, right, up, down and fire?**

Before going to great lengths to build such a device, don't forget that you can always use five keys on the Vic's normal keyboard (such as I,J,K,L and M), and use the GET statement to test for them. If you are keen, though, the figure here is a simple circuit which will allow you to read the keys through the user I/O port.

The keys can then be scanned by the Basic PEEK statement. For example:

```
100 POKE 37138,0: REM set
    VIA port B to input mode
110 I=PEEK(37136): REM get
    input data
120 IF (I AND 1) THEN PRINT
    "FIRE": GOTO 110
130 IF (I AND 2) THEN PRNT
    "DOWN": GOTO 110
140 IF (I AND 4) THEN PRINT
    "UP": GOTO 110
150 IF (I AND 8) THEN PRINT
    "RIGHT": GOTO 110
160 IF (I AND 16) THEN PRINT
    "LEFT": GOTO 110
```



Dear Tommy, **I have written a number of machine code subroutines for the Vic and would now like to incorporate them into Vic Basic as commands. Please could you explain to me how I may tokenise the keywords and arrange the pointers to point to my subroutine when one of the keywords is found by Basic.**

And how may I output a parameter, for example PRINT FRE (0), when one of my keywords is found.

There are several ways of doing this. The simplest way, which kills both your queries with one mixed metaphor, is to use the Basic USR function. You set the address of your subroutine in locations 1 and 2; and when you call the USR function the contents of the argument are left in the floating point accumulator. When you return from the function, the value you leave in the floating point accumulator is passed to Basic as the result of the function.

For example, if you say I – USR (5), the value 5 is left in the FAC for your routine to use (or to ignore, if you like). If you leave the number 423.265 in the FAC, then this is assigned to the variable I when you return.

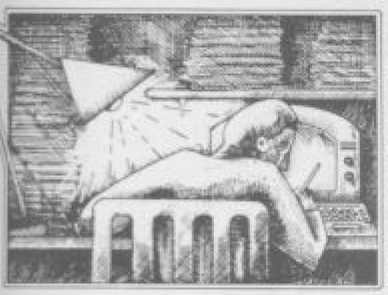
If you have too many subroutines, you may find changing the contents of locations 1 and 2 too boring. Then you are going to have to do a fair amount of work. There is not enough room in this column to give every detail, but hopefully I can point you in the right direction.

The easiest way is to patch CHRGET to jump to a routine of your own, by putting a JMP instruction at location $7C. This routine should preserve XR and YR, and you can then use TXTPTR ($7A and $7B) to examine text of the Basic program to see if you can identify one of your own keywords. If you find one, you can then jump to the relevant subroutine; if not you must carry out the normal CHRGET routine before returning.

As far as returning a value is concerned, you are going to have to assign to variables yourself – either by using one of the ROM subroutines or writing your own routine to search down the variable table.

Dear Tommy, **I am thinking about expanding my Vic. Can you tell me whether or not I could run a game designed for a VIC with a 3K expansion on one with a 16K expansion without any alteration? If so, what is the point of a switchable expansion?**

If the game has been properly written, it should check to see what size of Vic it is running in and make the necessary adjustments – though be warned that not all games do this. The problem is that the addresses of the Vic's screen and colour memory and the program area all move around when memory is



added. On the whole, it is probably safer to get a switchable expansion if you can.

Dear Tommy, **I am new to computing and I have come up with two problems which I hope you can answer. First, I understand that to get reversed letters I press CTRL and RVS ON. This is OK: but when trying to get reversed letters inside inverted commas it is not so easy – the reversed R keeps coming on. I have found that if I delete back to and including the inverted commas, retype the inverted commas and as quickly as possible type CTRL and RVS ON, I then get reversed letters as required.**

Is there a way round this without having to delete back to the start of the line?

The first question which springs to mind is "Why do you want to put reverse-video letters *inside* quotes?". What is wrong with a CTRL-RVS ON at the start? It does print the letters in reverse video, after all, which is the point of the exercise.

Anyway, the easiest way to get reverse characters inside quotes is to type "PRINT" and the first pair of quotes, then delete the quotes and type a quotes symbol again. You can then type "CTRL-RVS ON" followed by your letters – which will appear in reverse video.

The reason for this is that the Vic screen display works in two modes: normal and quotes mode. In normal mode, when you press a control sequence like CTRL-RVS ON, the character does not print on the screen; instead the control affects the following characters – in this case they appear in reverse video until you press CTRL-RVS OFF.

In quotes mode, pressing CTRL-RVS ON causes the RVS ON symbol to appear, and any following characters remain in normal video.

You change from one to the other by typing the quotes symbol ("); but deleting a quote symbol does not affect the display mode, so you can switch from one to the other by typing a quotes symbol and then deleting it.

Dear Tommy, **In the June 1982 issue you showed how to display a moving clock using machine code. But after NEWing the Basic loader there was more memory left than if there was no program at all. Is this real memory, or is the Vic fooling itself?**

I also have a number of musical questions:

1. Is it possible to enter two or more pieces of information into the Vic at the same time? I am trying to play music with more than just a simple melody. I have tried a combination of PEEK(197) and GET, but have not been too successful.

2. Is there a way of using alphanumeric keys on the keyboard to represent notes other than with a string of IF ... THEN statements?

3. Can the Vic be connected to proper speakers? And would it be possible to produce stereo?

The problem with the clock program is that line 100 should read POKE 56,29. Sorry about that!

As for your first musical query I presume you mean "Is it possible to detect that two keys are being pressed at the same time?". The answer is that it is possible in assembler, but not really practical in Basic. The best way to organise a program to play a piece of music in two parts would be to enter each part separately, and then play them together.

The IF-THEN alternative is feasible. Saying something like...

```
IF A$ = "A" THEN 1000
IF A$ = "B" THEN 1100
```

gets a bit boring if you have a large number of conditions to test for. Instead you can use ...

```
FOR I = 1 TO 10: IF A$ =
MID$("QFSS$6V(B£+",I,1)
THEN J=I:I=10
NEXT
ON I GOTO 1000, 1100, 1257,
3522 ...
```

... which can be a lot shorter. Obviously you will change the values to suit your own problem. If A$ can only take the values "A" to "G", say, running consecutively, you can simplify even more:

```
I = ASC (A$) - 64
ON I GOTO ...
```

The Vic can be connected to an amplifier driving the AUX input of the amplifier from the AUDIO output of the video/audio port. But if you are in any doubt on how to do it, you should consult your Commodore dealer. (*We've just commissioned an article on this — Ed.*)

Unfortunately you will not be able to get stereo, although you can drive both channels of a stereo amplifier in parallel.

Dear Tommy, **If I run the following program I get a screen of scrolling data:**

```
10 FOR I=1 TO 65535: PRINT
I,
   PEEK (I): NEXT
```

Where does the data for the contents of locations in the 40000 range come from when there is no ROM in the system to read?

If the program is run after power up, why does the RAM area contain variable data rather than one value? Memory contents aren't preserved during power off.

With a 6502 processor there is no 'handshake' between the processor and memory (as there is on the 68000, for example). To read a memory location, the 6502 just puts out an address; and a certain amount of time later, it reads the value on the data bus. If no memory is present at the specified address, this data will be indeterminate. You cannot call it 'random' in the sense of a random number generator because some numbers crop up more often than others, but it will be pretty meaningless.

On power-up, the value contained in memory locations are also interminate. Of course, large parts of the RAM are used by Basic (zero page, for example) or by the screen. So these are initialised before you are allowed to type in the program to examine them.

Dear Tommy, **When using text in conjunction with an INPUT statement it seems that the text must be less than 20 characters or it gets added to the beginning of the string being input. For example:**

```
100 INPUT "DO YOU WANT
    TO CONTINUE (Y/N)"; C$
```

If this line is run and "Y" is input, the resulting C$ string contains "DO YOU WANT TO CONTINUE (Y/N) Y" rather than just "Y". Is this a bug in the Basic?

Secondly, when a screen colour address is POKEd with a number (eg '2' for red) a subsequent PEEK at the location reveals a number which appears to be made up of the expected number added to a random number — either 32, 64, 160 or 224. Can you explain why this happens and the best way to overcome the problem?

First the INPUT problem. This is a well-known bug which appears in many Commodore machines, such as the early PETs, as well as the Vic. It is caused by the fact that when you print data over two lines, the Vic in effect considers them as a single long line. Unfortunately this confuses the INPUT statement so that it loses track of where the 'prompt' ends and where the data keyed by the user starts.

The second problem arises because those locations are in fact modified by the Vic as part of its screen handling. The simplest way out is to say . . .

```
X=PEEK(38400) AND 15
```

In this way X will only receive the colour information, with the other (higher) bits being masked out by the AND operator.

Two readers **have asked for help in using random numbers in assembly language programs.**

It is in fact quite simple to use the Vic's random number generator in the Basic ROM (used for the RND function). There are three entry points:

$E09B—corresponding to RND (0)
$E0BB—corresponding to RND (1)
$E0D0—corresponding to RND (−1)

The result is left in the floating point accumulator no.1 ($61-$66).

COMMODORE user

USE THIS SPACE
To promote your products to
Commodore 64 and Vic 20 enthusiasts
FOR AS LITTLE AS £30 A MONTH
Just ring Cathy Lane or Roland Richardson
on 01-241 2417
We'll be pleased to hear from you




BUSINESS ELECTRONICS LTD.
The Microcomputer Specialists
for
CBM ★ 64 ★ VIC20 ★ BBC MICRO ★ APPLE ★ SINCLAIR ★ SIRIUS
Micro-Magic in the Basement of Plummers
SOUTHAMPTON Telephone 0703-25903
or at
ROWNHAMS HOUSE, ROWNHAMS,
SOUTHAMPTON

Telephone: 0703-738248 or 739092 (Sales)

COLOUR MONITORS
PRINTERS — DISK UNITS
INTERFACES — CONNECTING CABLES
WORD PROCESSOR —
GAMES CARTRIDGES

CBM 64 in MILL HILL

ALL CBM — PET — VIC REPAIRS

H. NORMAN DAVIS LTD, 91 THE BROADWAY,
MILL HILL, LONDON NW7 3TG 01-959 2049

Sumlock

MANCHESTER.

Commodore appointed VIC 20 dealer.
Permanent showroom-shop display.
Hardware-Software-Accessories.
Books-Service-Maintenance.
Personal computers with Personal service.

198 DEANSGATE, MANCHESTER.
TELEPHONE 061 834 4233.

Everyman Computers

YOUR FRIENDLY
COMPUTER SHOP
IN WILTSHIRE



VIC PRINTERS
DISC DRIVES
SOFTWARE
DUST COVERS
BOOKS, CARTRIDGES

14 Edward Street,
Westbury, Wilts
BA13 3BD
(0373) 864644

Commodore 64-THE BEST VALUE ON THE MARKET

Specialists in microcomputers
Hardware & Software, systems &
programming, Equipment Rental

MICRO FACILITIES

Micro Facilities Ltd. 7-9 Church Road,
Egham, Surrey TW20 9QL Tel: Egham (0784 31333)

Y-E-S Vic 20 & 64

SALES & SERVICE

COMPUTERS

SAY YES FOR COMMODORE
HARDWARE, SOFTWARE & PERIPHERALS

YORKSHIRE
ELECTRONICS
SERVICES
LIMITED

Caxton House,
17, Fountain Street,
Morley, West Yorkshire
Tel: 0532-522181

COMPUTER CARRYING CASE

Suitable for Spectrum, Oric, Lynx, Sord, or computers of similar size. The hard case is finished in black with twin lockable fasteners and carrying handle.

Separate pockets hold computer, manual, leads, power pack, cassette player and five cassettes.

Folds flat for storage.

£13.95 inc VAT+
Post and packing £2

Dust covers also available for any computer, printer etc.

Dealer enquiries welcome.

SHERBORNE DESIGNS
Victory House, 8A The Rank,
North Bradley, Trowbridge, Wilts.
Tel: 02214 4425

---

ppm

PPM LIMITED...

For all your Business
Computer and
Software requirements.

We offer excellent after sales support including a team of experienced service engineers.

To discuss your application contact:
David Chester on: 04867 80111

Hermitage Road, St Johns, Woking, Surrey GU21 1TZ
Telex: 850181

---

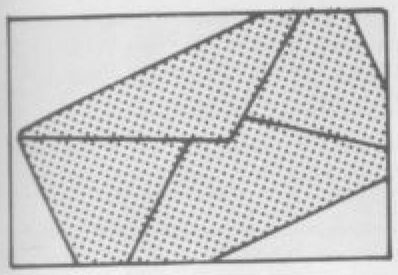
fbc

DERBY

CBM64 AND VIC20 DEALERS

We have the largest selection
of hardware, software and books
in the area.

FIRST BYTE COMPUTERS,
10 MAIN CENTRE,
LONDON ROAD, DERBY.
TEL: 0332 365280



On advertisers and suppliers...

I am writing to your magazine to indicate how pleased I am with the service which I have received from one of your advertisers.

During the afternoon of Monday 18 July I rang Stack Computer Services Limited. I wished to purchase their four-slot mother board for my Vic 20. The girl stated that the boards were in stock. I asked how long I could expect delivery to take. She said about two days. I ordered the board then and arranged payment.

At 3.15pm on Tuesday 19 July the post van arrived outside my home and to my amazement my mother board had arrived. Five minutes later it was installed and working.

If this is typical of the service offered by this company they deserve to be commended.

Clive G. Elesmore, 52 Silverlands Road, Lyminge, Folkstone, Kent CT18 8JG.

Morse Trainer

I refer to the program *Morse Trainer* which appeared in the June issue of **Vic Computing**. Clarence Buckley's problem of running out of memory is due to the fact that when GOSUB 730 is executed, the program goes "merry-around-a-gosub" until the stack is full, whence the **OUT OF MEMORY ERROR** is displayed.

To overcome this problem, replace line 610 with:

610 PRINT "[CUD]" : GOTO 150

...and delete line 730. Having done this, the number 730 in lines 560 to 600 is redundant, and can be deleted. Similarly, the statement F = 4 in lines 160 to 150 is redundant, and can also be deleted (F is *always* 4; and A, B, C, D, or E is *never* 4).

To restrict the action to the alpha and numeric keys you can insert a condition statement between lines 540 and 550:

IF M$ = CHR$ (X) THEN 150

...where X = CHR$ value of the key to be excluded. For example, if the following lines are inserted:

542 IF M$ = CHR$ (13) THEN 150
543 IF M$ = CHR$ (19) THEN 150
544 IF M$ = CHR$ (20) THEN 150
545 IF M$ = CHR$ (29) THEN 150
546 IF M$ = CHR$ (32) THEN 150

... then pressing the RETURN, CLR/HOME, INST/DEL, CRSR RIGHT and SPACE keys will not repeat the last sent signal.

And finally, to make the input of speed robust, insert a line 105:

105 IF R$ <> "F" AND R$ <> "M" AND R$ <> "S" THEN 70

Mohamed Noor, 48 Edgeware Rd, Singapore 2879

Mother Hen

I would like to point out a slight bug in the **Mother Hen** program in the February issue.

The program contains several lines which check that the data is correct in the first program Charset. However, even if the data is correct the program still comes up with DATA ERROR. This can be rectified by changing line 1020 to read:

1020 READ B : IF B > 256 THEN 1060.

C. Pickin, Levens, Tunstall Road, Knypersley, Stoke-on-Trent, ST8 7AB

Bad news

I read with interest your article on DAMS Vic add-ons. Perhaps I can add something else for my fellow readers to think about.

I ordered a light pen in September 1982. It arrived faulty in December and was returned. A subsequent one with the same fault was returned to DAMS in January on the promise of a refund from them. Six months later I was still waiting for the refund despite phone calls every week for the first five months and every two working days since.

They are always very courteous and promise to 'get it sorted' but nothing ever happens.

I have been unable to contact the Company Secretary for the last three weeks and apparently noone else can sign a cheque! I don't know whether this is typical of their after-sales service or just their reluctance to part with the cash. Have any of your other readers had any problems?

I am very impressed with the new format of the magazine particularly as I now own a Commodore 64 as well as the Vic.

Incidentally I do hope the much vaunted Simon's Basic cartridge for the 64 doesn't go the same way as the always-promised but never-seen (at least by me) 40 column card for the Vic 20...

B.D. Parsons, 15 Maplewood Grove, Saughall, Chester, CH1 6AD.

He's back again...

I recently received a copy of your new magazine, and I thought you might like to know what I think.

1 Full marks for the contents printed on the first page and in a logical manner by page number. Too many magazines divide contents into parts like 'Features', 'Listings' etc. This is very annoying! Also in earlier issues of Vic Computing the contents pointed to the wrong pages - no problems here.

2 I found the white on black printing of the Editorial hard to read – please do not do this in future.

3 I'm British and therefore cannot think in dollars. How about putting the £ equivalent in brackets?

4 I read with interest the article on 'Making money from your hobby'. I am 14 and selling programs to the company Mr. Micro Ltd. who give me 20% royalties. So teenagers, do not give up hope... Also Basic is not necessarily out – I write most of my games half in Basic and a half-and-a-bit in machine code, though obviously this varies.

5 I was annoyed with the quote in the 'Filing' article that said: 'In practice most computers don't do an awful lot of computing – not in the sense of performing mathematical calculations, anyway'. Absolute bunk. All computers fundamentally work with the three logical operators AND, OR, NOT – and if that isn't maths, what is? *(It's argueable, that's what it is. Or semantics, anyway. Most computers shuffle characters and decode signals using mathematical techniques; but they aren't usually operating directly on numbers. Ed.)*

6 I found the article on 'Assembly language programming confusing to say the least – and I thought I already knew binary, hex, and machine code. *(Funny, other people have said the same things about it. Ed.)*

7 In the article 'Machine code vs Basic', you stated that the cassette buffer started at 828 and held 192 bytes maximum. Actually, you can use 204 bytes with the locations 820-1023.

8 On page 31 I was puzzled by the two lines 100 until I realized that the first was a misprint. It should read **100 X = X = (Y = 15)**. I defend this method because instead of saying...

100 P = PEEK (197): IF P = 50 THEN A = A − 22
110 IF P = 35 THEN A = A + 22
120 IF P = 42 THEN A = A − 1
130 IF P = 43 THEN A = A + 1

... you can use...

100 P = PEEK (197) A = A + 22 * ((P = 50) − (P = 35)) + (P = 42) − (P = 43)

... or even...

10 DEF FNM (X) = 22+ ((X = 50) − (X = 35)) + (X = 42) − (X = 43)
100 P = PEEK (197) IF P = 32 THEN GOSUB 200 REM fire button
110 A = A + FNM (P)

9 I am opposed to 'computer holidays'. For a start their brochures say "programme" where they mean "program". And how much will you learn in a week? I am not blaming you, you just review the things, but I would like to make my opinion known. *(Gosh thanks. Ed.)*

10 The advert on page 75 said 'software supplied to save cartridges to tape or disk'. Breach of copyright? Remember, unless the contract with the advertiser states otherwise, you are responsible for all the material in the magazine, and any cost or damages this would imply. (I'm not very good at legalese, am I?) *(Not so bad, Ian. Ed.)*

11 In the reviews of programmer's aids you list commands. All very well – but what is the **CLEAR** command? It doesn't exist on my programmer's aid... (Also, on mine the **MERGE** facility never works. I get **?MERGE ERROR**. Do other people have this problem? Can it be cured?)

Can I recommend the book *6502 Assembler* by Lance A. Leventhal? It contains a complete description of the instruction set – and I mean *complete*; a couple of A5 pages on each instruction. Also there are lots and lots of programs which if not useful in themselves are certainly an exercise in programming technique.

But what I fell in love with in this book was the printing. Important things are in boldface type, and lesser things or an expansion on the previous boldface type is in lightface type. So you can read the bold type until you reach a subject that you want to know more about, at which point you read the light type.

Also the contents pages – all 14 of them – are a boon. Chapters are subdivided on the spot so you find what you want straight away. Not a book for the beginner, but I have found it an invaluable reference.

Ian Creasey, 182 Didsbury Road, Heaton Mersey, Stockport, Cheshire SK4 2AA.

A bagful of ideas...

Here are some ideas which you can file in that round filing cabinet under your desk.

1: How about starting a software exchange scheme among readers? I have about 20 programs that I will swap!

2: Why not do a round-up of all the different joysticks available? Some time ago I bought a Stack analogue joystick – if any software uses it I haven't seen it.

3: How about a second-hand equipment corner – anyone want to buy my Programmers Aid Cartridge? (I'm trying to raise enough money for a 16K RAMpack, you see.)

4: We've all complained about the inadequacies of the present-day computers – the Vic's small screen, the ZX81's horrible excuse for a keyboard, etc. So why not a competition to design the perfect micro?

5: I think you should do a series of articles on machine code. I would volunteer, except that I am not a very good writer. So the next best thing is a program, and here's a program you will be asked to assign strings to 12 keys: F1, F3, F5 and F7; F2, F4, F6 and F8 (with SHIFT); and F9, F10, F11, F12 (with the CBM key). If you want a carriage return in a string, use a left arrow. Now you can NEW the program (after saving on tape, of course) and the function keys will work. To stop them working, hit RUN/STOP and RESTORE; to start again, type SYS755.

Here is a disassembled listing of the interrupt wedge:

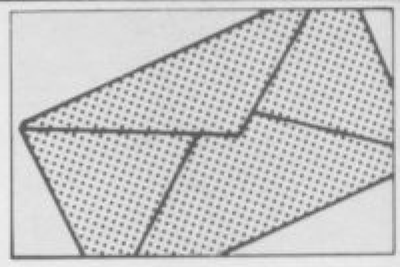
```
        LDA 197      ; Get key
        CMP 0        ; Same as old key?
        BEQ EXIT     ; If yes, exit
        STA 0        ; Store new key
        LDX # 1      ; Value of key 1
        CMP # 39     ; Is it key 1?
        BEQ SHIFT    ; If yes, check for shift keys
        INX          ; Value of key 2
        CMP # 47     ; Is it key 2?
        BEQ SHIFT    ; If yes; check for shift keys
        INX          ; Value of key 3
        CMP # 55     ; Is it key 3?
        BEQ SHIFT    ; If yes, check for shift keys
        INX          ; Value of key 4
        CMP # 63     ; Is it key 4?
        BNE EXIT     ; If not, exit
SHIFT   LDA 653      ; Get shift value
        AND # 3      ; Extract relevant bits
        BEQ STRING   ; If no shift keys, get string
        CMP # 3      ; Both pressed?
        BEQ EXIT     ; If yes, exit
        INX          ; Add 4
        INX          ; for
        INX          ; SHIFT
        INX          ; key
        CMP # 1      ; Check for SHIFT key
        BEQ STRING   ; If yes, get string
        INX          ; Add another 4
        INX          ; for
        INX          ; CBM
        INX          ; key
STRING  LDY # 0      ; Set index to zero
CHECK   DEX          ; Check whether reached right key
        BEQ BUFFER   ; If yes, store string in buffer
        INY          ; Increment index
GET     LDA 819, Y   ; Get character
        BEQ CHECK    ; If end of string, check whether reached right key yet
        BNE GET      ; Get another character
BUFFER  LDA 820, Y   ; Get character
        BEQ NUMBER   ; If end of string, set pointer
        STA 631, X   ; Store character in buffer
        INY          ; Increment
        INY          ; Indices
        BNE BUFFER   ; Get another character
NUMBER  STX 198      ; Set keyboard buffer pointer
EXIT    JMP 60095    ; Jump to normal interrupt
```

As this program resides in the free space from 673 to to 765, it doesn't take up any of your precious Basic memory.

I hope this listing will prove useful – when I was learning machine-code I felt a great need for simply explained examples of real programs.

Have you ever wanted to 'remember' the X,Y coordinates of the location where the next PRINT will occur in your Vic-20 programs? Have you ever wanted to be able to set the coordinates back again after having printed in another part of the screen? The following functions and subroutine will allow you to do this:

```
10 DEF FNA(X)=PEEK
   (214)* 22+PEEK (211)
20 DEF FNX(A)=A-INT
   (A/22)* 22
30 DEF FNY(A)=INT
   (A/22)-1
100 REM your program
110 PRINT"[clear][10 down]
    [10 right]";
120 X=FNA(0)
130 PRINT"[3 right]
    [3 down]*"
140 GOSUB 1000 :
    PRINT"I":END
1000 POKE 214, FNY(X):
     PRINT: POKE211,
     FNX(X): RETURN
```



When run, the '*' will print under the 'I', where it would have printed earlier, just like you wanted it to!

Scott C. Jensen, 1359 Palace Avenue, St. Paul, MN 55105, USA

I am interested in hearing from people who have written practical business application programs for the Vic-20.

It is my firm opinion that this small system has many practical business applications, and I am forming a software sales organization for the purpose of encouraging practical business use of the system.

I will be pleased to hear from anyone who has developed such programs, and in all probability it will be to their financial advantage to contact me.

A E Danies, PO Box 1854, Ojai, CA 93023, USA

Ad Index

“robustness and reliability were excellent... a really good feel to the action... a pleasure to use... not one game failed to load... nice professional ... I like it... very impressed”

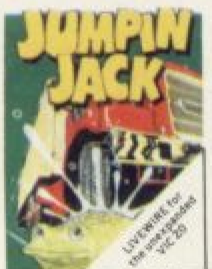
EXTRACTS: J.D. COLLINS
GIANT TEST VIC COMPUTING
(Vol. 2 Issue 5 June, '83)

DON'T JUST TAKE OUR WORD FOR IT!

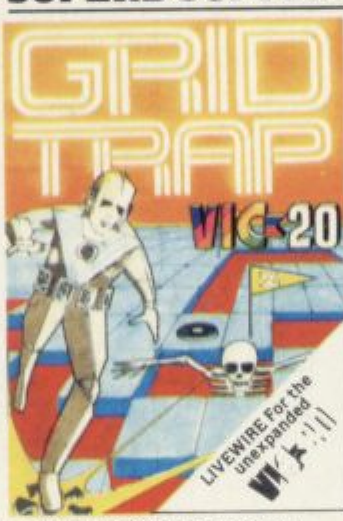
SUMLOCK MICROWARE
SUPERB SOFTWARE FOR THE VIC 20



SP.01 SKRAMBLE Destroy the enemy installations and missiles in their underground silo. Joystick or keyboard. M/code. **£7.95**



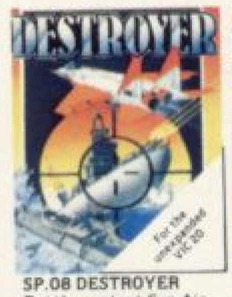
SP.02 JUMPIN JACK Cross a busy road and treacherous river to get Jack home. Joystick or keyboard. Multi-colour Hi-Res Gr. M/code. **£7.95**



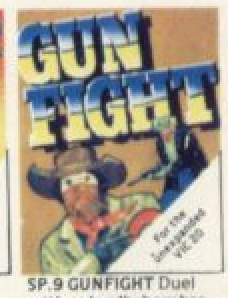
SP.04 GRIDTRAP Your man has been placed on a grid of traps which open after he walks over them. Defuse the time bombs in less than 60 seconds. Further hazards are presented by a series of mines scattered around the grid which cannot be walked on and a wandering 'BOOT' which must be avoided. The more you play this one the harder it gets, bonus life 100,000 points. Programmed in Machine Code. Hi-res Colour Graphics and novel musical sound effects. Joystick or keyboard control. **£7.95**



SP.16 MULTITRON Lightning responses are demanded when manning your Starfighter through what can only be described as the biggest alien encounter of all time. Use your laser cannon, destroy or be destroyed. The Cosmic Phoenix, Space Turtles, Star Hoppers, Tri-Missiles, Warp Stingers, and Meanies defend the space lanes in wave after wave of invasion action... The ultimate journey now awaits you... The winding space corridor! Hi-res colour graphics. Sound effects. Joystick or keyboard. M/code. **£7.95**



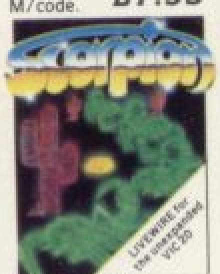
SP.08 DESTROYER Battle against Sea Air attack with the destroyer H.M.S. Victory. Joystick not required. M/code. **£7.95**



SP.9 GUNFIGHT Duel with a deadly hombre in a western shootout play the computer or 2nd player. Joystick or keyboard. M/code. **£6.95**



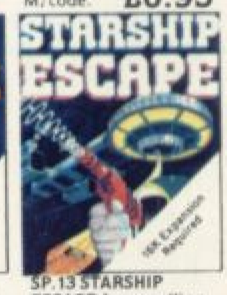
SP.03 TRIAD Defend your base against the suicidle fleets of alien hoverships. Joysticks or keyboard. M/code. **£7.95**



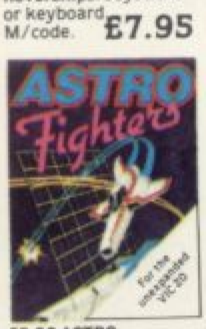
SP.15 SCORPION Destroy the poisonous cacti and deadly scorpions to survive. Joystick or keyboard. M/code. **£7.95**



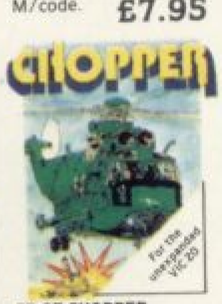
SP.10 SPACE RESCUE Pilot your rescue craft through the asteroid belt and save the stranded crew. Joystick or keyboard. M/code. **£6.95**



SP.13 STARSHIP ESCAPE A compelling and exciting real time graphic space adventure game. Joystick or keyboard. M/code. **£9.95**



SP.06 ASTRO FIGHTERS A space duel played by two players at once. Requires Joystick. M/code. **£6.95**



SP.07 CHOPPER Attack and destroy the enemy base but watch out for guided missiles. Joystick or keyboard. M/code. **£6.95**

... AND THE COMMODORE 64

LW01 SKRAMBLE Enhanced features. Hi-res graphics. Sprite graphics. Synthysound. No. 2 option. Pixel plot graphics. Screen scroll. **£8.95**

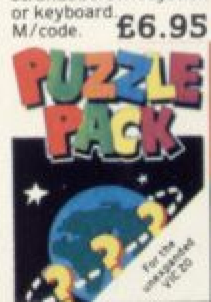
LW02 JUMPIN JACK Enhanced features. Hi-res multi-colour graphics. Sprite graphics. Synthysound. 2 Player option. Lady Frog. Crocodiles etc. **£8.95**

L03 TRIAD Hi-res graphics. Sprite graphics. Synthysound. Pixel plot graphics. 2 Player and game level option. **£8.95**

LW04 GRIDTRAP Hi-res graphics. Sprite graphics. Synthysound etc. **£8.95**

SP.17 AT THE CIRCUS Requires 3K or more expansion Educational adding and subtracting program for 4-8 year olds 3 Levels animated graphics. (not illustrated) **£5.95**

SEND FOR YOUR SUMLOCK MICROWARE NOW OR FOR A FREE BROCHURE AND YOUR NEAREST DEALER INFORMATION



SP.12 PUZZLE PACK A compendium of six intriguing puzzles and games for all the family. **£6.95**



SP.14 DATA PACK A three program data file handling utility package and combined educational quiz. **£8.95**

AMERICAN EXPRESS
Access
BARCLAYCARD VISA

061 834 4233

SUMLOCK MICROWARE
198 Deansgate, Manchester, M3 3NE.

AVAILABLE FROM GOOD COMPUTER DEALERS AND BOOK SHOPS NATIONWIDE INCLUDING: WILDINGS, SHERRAT & HUGHES, JOHN MENZIES. OVERSEAS EXPORT ORDERS ON APPLICATION. ALL SOFTWARE IS FULLY GUARANTEED. PRICES INCLUDE POSTAGE & PACKING.

PLEASE SEND ME THE FOLLOWING MICROWARE (QUOTE Nos.)

I ENCLOSE CHEQUE/P.O. £

TO. NAME

ADDRESS

DPT. NO. VCS1